T.C.
MİLLÎ EĞİTİM BAKANLIĞI

YENİLENEBİLİR ENERJİ TEKNOLOJİLERİ

SERİ PORT KONTROLÜ

Ankara, 2015

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- PARA İLE SATILMAZ.

İÇİNDEKİLER

AÇIKLAMALAR ii
GİRİŞ 1
ÖĞRENME FAALİYETİ-1 3
1. SERİ PORTTAN ÇIKIŞ ALINMASI 3
1.1. Seri Haberleşme Çeşitleri 4
1.1.1. Eş Zamanlı Haberleşme(Synchronous) 4
1.1.2. Eş Zamansız Haberleşme(Asynchronous) 5
1.2. Sinyal Seviyeleri 5
1.3. Seri Port Adresleri 6
1.4. Seri Port Bağlacları (Konnektörü) 7
1.5. Seri Port Programlama 9
1.6. Dil Dosyası ile Seri Porta Erişim 9
1.7. API Fonksiyonları ile Seri Porta Erişim 10
1.8. Seri Porttan Karakter İletimi 27
UYGULAMA FAALİYETİ 34
ÖLÇME VE DEĞERLENDİRME 35
ÖĞRENME FAALİYETİ–2 36
2. SERİ PORTA GİRİŞ YAPILMASI 36
2.1. Seri Porttan Sinyal Okuma 36
2.2. MsComm Bileşeni 45
UYGULAMA FAALİYETİ 48
ÖLÇME VE DEĞERLENDİRME 49
ÖĞRENME FAALİYETİ–3 50
3. İKİ BİLGİSAYARI BAĞLAMAK 50
3.1. Seri Haberleşme Standartları 51
3.1.1. Bağlantı Şekilleri 51
UYGULAMA FAALİYETİ 63
ÖLÇME VE DEĞERLENDİRME 64
MODÜL DEĞERLENDİRME 65
CEVAP ANAHTARLARI 66
KAYNAKÇA 67

AÇIKLAMALAR

ALAN	**Yenilenebilir Enerji Teknolojisi**
DAL/MESLEK	**Bilgisayralı Kontrol**
MODÜLÜN ADI	**Seri Port Kontrolü**
MODÜLÜN TANIMI	Seri portun yapısı ve visual basic programlama dilinde seri port üzerinden veri alış verişi işlemlerini yapabilme yeterliliğinin kazandırıldığı modüldür.
SÜRE	40/32
ÖN KOŞUL	
YETERLİK	Seri port kontrolü yapmak.
MODÜLÜN AMACI	**Genel Amaç:** Seri port ile giriş ve çıkış işlemlerini doğru olarak yapabileceksiniz. **Amaçlar** **1.** Seri port kontrol devresini devre şemasına va baskı devre tekniklerine uygun olarak yapabileceksiniz. **2.** Seri port donanımı yoluyla sayısal giriş değerlerini kontrol edebileceksiniz. **3.** İki bilgisayar arasında veri iletişimini sağlayabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	**Ortam:** Bilgisayar Laboratuarı, Elektrik-Elektronik Laboratuarı **Donanım:** Visual Basic 6.0 veya Microsoft Visual Basic 2010 Express çalıştırabilen bilgisayar, bilgisayar çevre birimleri, lehimlelme araç gereçleri, multimetre, güç kaynağı **Yazılım:** Visual Basic 6.0. veya Microsoft Visual Basic 2010 Express programlama dili
ÖLÇME VE DEĞERLENDİRME	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyiniz ölçeceksiniz. Araştırmalar, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz.

GİRİŞ

Sevgili Öğrenci,

Dos tabanlı programlama dillerinin tahtına oturan görsel diller, denetlenen donanımla kullanıcının kolay ve anlaşılabilir bir arayüz kurmasını sağlamıştır. Donanımla ilişkili programlar yazmak hiç bu kadar zevkli olmamıştı. En basitinden bir düğmenin Dos tabanlı programlarda yazılması, üzerinde fare ile tıklatılarak bir şeyler yapılmasının zahmeti düşünülecek olursa şimdiler de günlük hayatımız değilse bile programlama hayatımızın bir hayli kolaylaştığı söylenebilir.

Bu modül üç bölümden oluşmaktadır.

- Seri portlarıdan çıkış alınması
- Seri portlara giriş yapılması
- Seri portlarla haberleşme

Öğrenme faaliyetlerinde konu genel olarak değil örnekler üzerinde anlatılmıştır. Bu yöntem, yapacağınız uygulamalara rehberlik edecektir. Örneklerde, yazılacak programın ekran görüntüsü, örnek program ve açıklamaları verilmiştir. Bu yol ile uygulamalarda yararlanacağınız işlem basamaklarını daha iyi anlayacaksınız.

ÖĞRENME FAALİYETİ-1

AMAÇ

Seri port kontrol devresini devre şemasına va baskı devre tekniklerine uygun olarak yapabileceksiniz.

ARAŞTIRMA

- Endüstride, okulunuzdaki atölye ve laboratuarlarda seri port kullanarak haberleşme yapan cihaz ya da sistemleri araştırınız.

1. SERİ PORTTAN ÇIKIŞ ALINMASI

Seri port endüstriyel uygulamalarda kullanıldığından dolayı masaüstü bilgi-sayarlarda ve dizüstü bilgisayarın hemen hemen hiç birinde seri port ve paralel port mevcut değildir. USB çıkışını seri porta çıkışına dönüştüren çeviriciler bulunmak-tadır. Endüstride yaygın olarak kullanılan seri port D tipli 9 pinli seri porttur ve ismi RS232 olarak geçmektedir. USB to RS232 dönüştürücü ile USB çıkışı seri porta çevrilir.

Resim 1.1: USB'yi seri porta çeviren dönüştürücü

Bir paralel port 8 bitlik veriyi 8 ayrı kablo üzerinden alır veya gönderir. Bu verinin daha hızlı aktarımını sağlarken kullanılacak veri kablosu sayısını arttır-maktadır yrıca 5 volt gerilimle çalıştığı için veri uzak mesafelere aktarılırken sin-yal zayıflamakta verinin ne

olduğunun algılanması zorlaşmaktadır. Paralel port için tavsiye edilen azami uzunluk 6 m denilse de bu gerçekte kablo durumuna göre değişmektedir.

Seri port veriyi tek bir kablo üzerinden iletir, diğer bir kablo üzerinden alır. Bu, 1 baytlık bilginin 8 bit ardışık olarak iletilmesi anlamına gelmesine rağmen daha az kablonun da kullanılması anlamına gelmektedir.

Seri port, RS-232C standardına uyumludur. 1969 yılında geliştirilen bu standart, seri iletişimin temellerini belirlemiştir. Daha bilgisayar üzerinde seri port yokken bile seri iletişim kuralları tesbit edilmişti. RS-232C (Recommended Standard) ismi 232 no'lu tavsiye edilen standart anlamındadır. C ise standardın son gözden geçirilmiş hâlini belirtir.

Seri iletişim ilk olarak telefon hatlarını haberleşme hattı olarak kullanan elektronik daktilolarda (teletype machines) kullanılmıştır.

1.1. Seri Haberleşme Çeşitleri

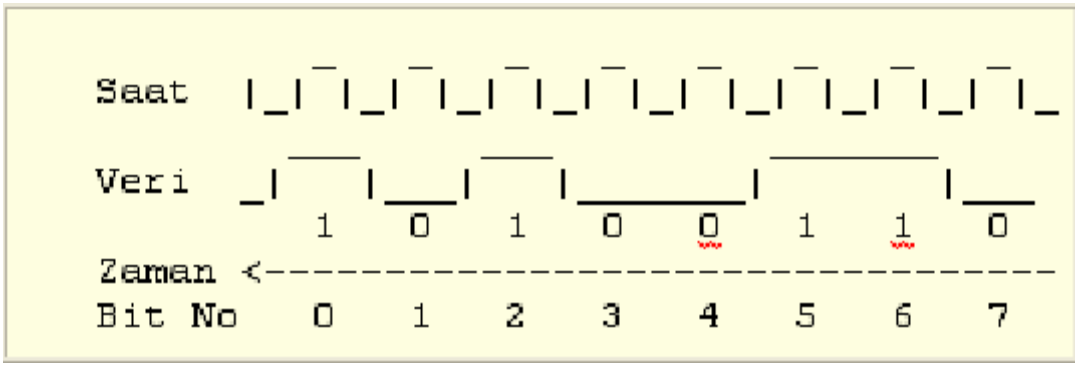
Seri haberleşmede, her iki uçta yer alan aygıtların birisine DTE(Data Terminal Equipment: Veri Uçbirim Cihazı), diğerine de DCE(Data Communications Equipment: Veri Haberleşme Cihazı) denilmektedir.

Bilgisayar DTE'ye modem ise DCE'ye birer örnektir. Bu şekilde karşılıklı haberleşen aygıtların birbiriyle uyum içerisinde olması gerekir. Bilgi nasıl taşınacağı, karşıdakinin bundan nasıl haberdar olacağı gibi konuların önceden belirlenmesi gerekir. Bu duruma göre seri haberleşme ikiye ayrılmaktadır.

Bu işlemler sonucunda aşağıdaki şekilde görüldüğü gibi karşımıza onaltılık sayı tabanında adresleme tablosu gelir. Farklı bir bilgisayarda aynı tabloyu görmeye çalıştığımızda adres sıralarının boş ya da sırasının değişik olduğunu görebiliriz. Diğer bir noktada bilgisayarımızda tek bir tane seri port olmasına karşın bu tabloda üç tane adresin var olduğunu görebiliriz. Bu durumlar BIOS versiyonlarından kaynaklanır.

1.1.1. Eş Zamanlı Haberleşme(Synchronous)

Veri bitlerine eşlik eden bir saat sinyali vardır. Bu sinyal ya alıcı ya da verici tarafından üretilir.

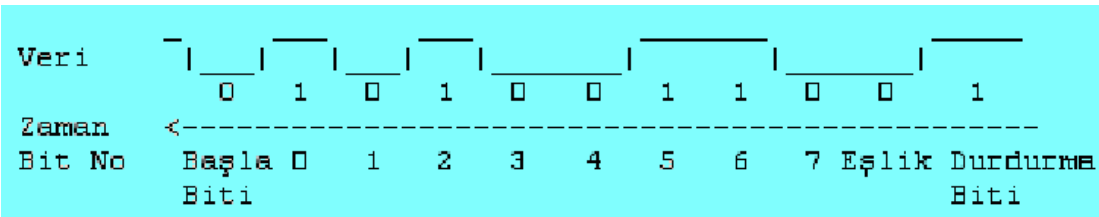


Şekil 1.1: Eş zamanlı haberleşme

Her iki aygıt başlangıçta birbirini haberdar ettikten sonra veriler gönde-rilir. Veri gönderilmediği anlarda bile boş karakterler, gönderilerek hatta kalınması temin edilir.

1.1.2. Eş Zamansız Haberleşme(Asynchronous)

Alıcı ve verici aygıtlar aktarım hızı(baud hızı), veri paketinde bulunacak bit sayısı gibi konusunda önceden anlaşır. Veriler gönderilmeye başlamadan önce bir başlangıç biti, aktarım bittikten sonra da bir durdurma biti gönderilir.



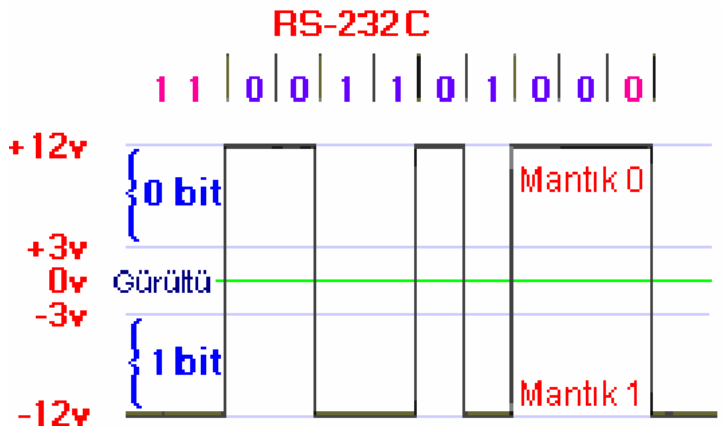
Şekil 1.2: Eş zamansız haberleşme

Başlangıç ve durdurma bitlerinin veriye eklenmesi dolayısıyla bu haberleşme eş zamanlı haberleşmeye göre daha yavaştır. Seri port eş zamansız haberleşme yapmaktadır.

1.2. Sinyal Seviyeleri

Seri port, -12V / +12V aralığında çalışmaktadır. Böylelikle veri daha uzun mesafelere taşınmaktadır.

Seri port için toprak hattına göre -12V gerilimine “Mantık 1”, 12V gerilimine ise “Mantık 0” seviyesini ifade eder. Mantık 1 seviyesine, “Mark” (iz), Mantık 0 seviyesine “Space” (boşluk) da denilmektedir. Standarda göre -3V ve -30V arası “Mantık 1”, +3V ve +30V arası “Mantık 0” dır. -3V ve +3V arasındaki bir gerilim belirsizdir. Çıkış akım yaklaşık 10mA civarındadır. Diz üstü bilgisayarlarda bu gerilim seviyesi -7.5V ve +7.5V arasında değişmektedir.



Şekil 1.3: Seri port sinyal seviyesi

1.3. Seri Port Adresleri

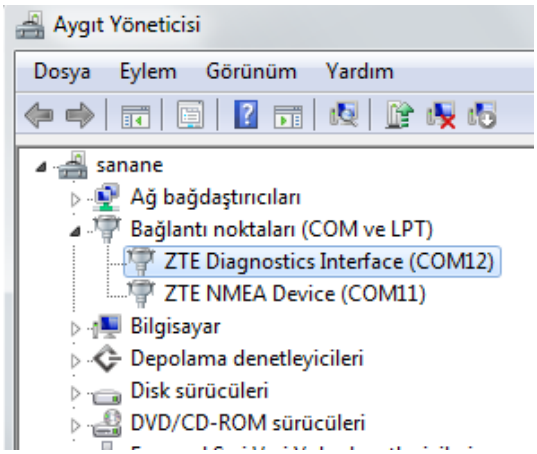
Bilgisayarda seri haberleşme, UART isimli yonga tarafından gerçekleş-tirilmektedir. Atası 8250 olan bu yonga zaman içerisinde 16450, 16550, 16650 ve 16750 numaralarını alan UART (Universal Asynchronous Receiver / Trans-mitter), çok yönlü eş zamanlı gönderici / alıcı anlamındadır.

UART'a bağlı olan seri port COM1. COM4 olarak adlandırılır ve bilgisayar üzerinde Tablo 1.1'de görülen adreslerde bulunabilir.

Seri Port	Adres (hex)	Adres(onluk)
COM 1	3F8	1016
COM 2	2F8	760
COM 3	3E8	1000
COM 4	2E8	744

Tablo 1.1: Seri port adresleri

Eğer USB – RS232 dönüştürücü kullanıyorsanız, seri port adresini (Com1, Com2, ... Com 15 vb) bilgisayarınızın aygıt yönetici penceresinden görebilirsiniz.

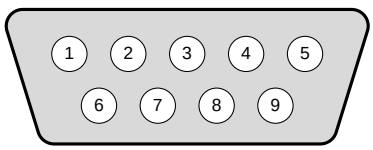


Resim 1.2: Bilgisayar bağlantı noktaları

1.4. Seri Port Bağlacları (Konnektörü)

RS-232C standardı, 22 tanesi kullanılan 25 pinli D tipi bir konnektörü tanımlamaktadır. Bunların bir çoğu normal bilgisayar haberleşmelerinde kullanılma-dığından IBM, bu pinlerin sayısını 9 olarak belirlemiştir. Yeni bilgisayarların hepsi D tipi erkek bir seri port bağlacı ile gelmektedir.

Seri port üzerinde bulunan pinlerin anlamları Tablo 1.2'de görülmektedir:

RS232 konnektörü	
Pin Numarası	**Sinyal Tanımı**
1	**Carrier Detect (CD):** modem tarafından "Ben başka bir modeme bağlandım."demek için kullanılır.
2	**Received Data (RxD):** Bilgilerin alındığı uçtur. Gerilim 0V civarındadır.
3	**Transmitted Data (TxD):** Bilgilerin gönderil-diği uçtur. Veri gönderilmediği durumlarda geri-limi –12 V (Mantık 1)'dir.
4	**Data Terminal Ready (DTR):** DSR pinsi ile ortaklaşa kullanılır. İşlev olarak RTS-CTS ile aynı işi, yani donanım el sıkışmasını yapar. Yalnızca sorduğu soru farklıdır. Bu hattı kullanarak bilgisayar, porta bağlı diğer bir aygıta "Sana enerji verildi mi? Bana bağlı mısın? Çalışır vaziyette misin?" sorularını sorar. Bilgisayar ilk açıldığında –12V (Mark ya da

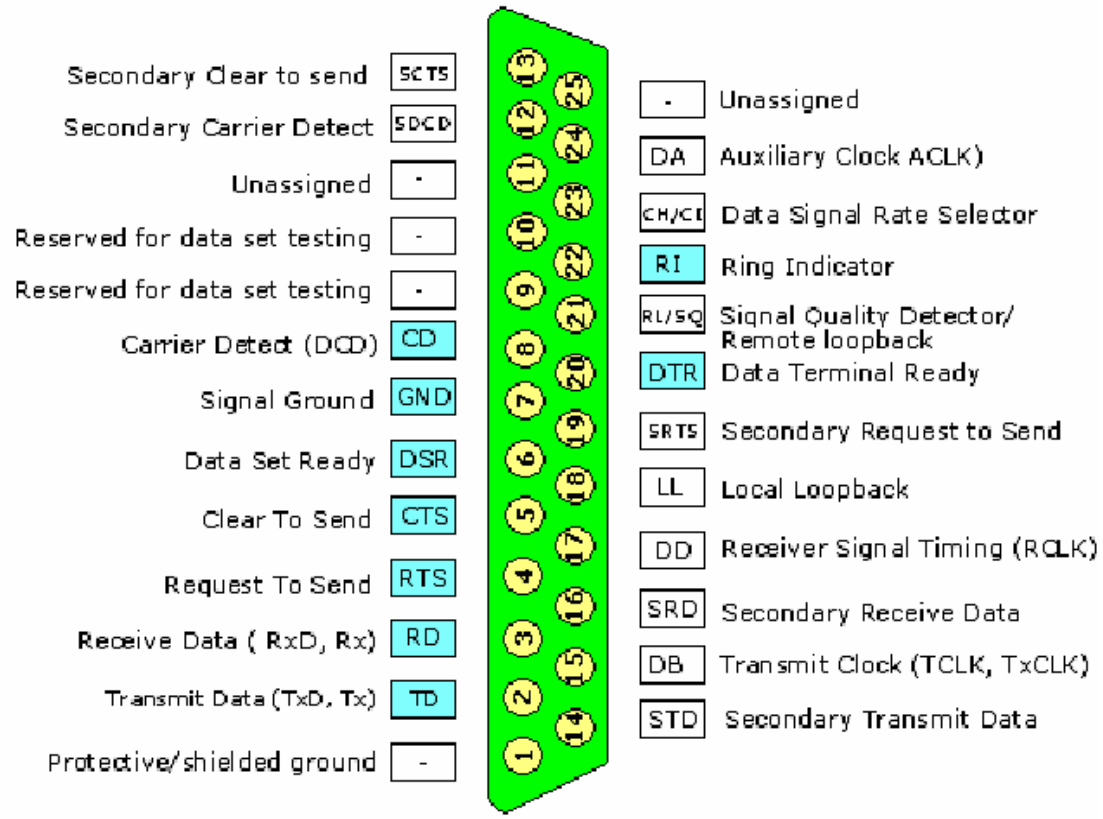
	Mantık 1) seviyesindedir.
5	**Signal Ground: Haberleşme** yapılacak aygıtın toprak hattı ile birleştirilmelidir.
6	**Data Set Ready (DSR):** Harici aygıt bilgisayara bu hat aracılığıyla "Evet. Enerjim var. Hazırım." der.
7	**Request To Send (RTS):** Çıkış ucudur. Bilgisayar ilk açıldığında –12V (Mark ya da Mantık 1) seviyesindedir. CTS ile beraber donanım el sıkışması için DTE bu hattı, "Bir karakter göndermek istiyorum. Hazır mısın?" sorusunu DCE'ye sormak için kullanır. Her bir karakter gönderimden önce bu hat "Mantık 1" seviyesine geti-rilerek bu soru sorulur.
8	**Clear To Send (CTS):** Giriş ucudur. DCE bu soruya "Evet, hazırım. Karakteri gönder" demek için bu hattı kullanır. Bunun için bu hattı mark durumuna getirir. Alamayacak durumda ise space durumundadır.
9	**Ring Indicator (RI):** Modem bu ucun durumunu bir telefon zil sesi geldiğinde sürekli olarak değiştirir.

Tablo 1.2: Seri port pinleri

Tablo 1.3'te 9 uçlu konnektörün 25 uçlu konnektördeki karşılıkları görül-mektedir.

9 lu	İsim	Giriş/Çıkış	25 li
3	TxD	Çıkış	2
2	RxD	Giriş	3
5	GND	-	7
7	RTS	Çıkış	4
4	DTR	Çıkış	20
8	CTS	Giriş	5
6	DSR	Giriş	6
1	DCD	Giriş	8
9	RI	Giriş	22

Tablo 1.3: Eş değerlik tablosu



Şekil 1.4: 25'li seri port iğneleri

1.5. Seri Port Programlama

Seri portun programlanmasında üç farklı yol takip edilebilir. Bunlar:

- Paralel ve seri porta erişim için yazılmış bir dil dosyasını kullanarak,
- Seri port API fonksiyonlarını kullanarak,
- VB'nin seri port için yazılmış MsComm bileşenini kullanarak gerçekleşir.

Bu yöntemleri ayrı konu başlıklarında işleyeceğiz.

1.6. Dil Dosyası ile Seri Porta Erişim

Paralel ve seri porta erişim için, C/Visual C veya Pascal / Delphi prog-ramlama dillerinde yazılmış bir dll dosyası, Windows'un system klasörüne kop-yalanmalıdır. Bizim burada kullanacağımız dosya, "inpout32.dll" dosyasını,
http://www.logix4u.net/inpout32.htm adresinden indirebilirsiniz.

Yalnız bu dil dosyası, Windows 98 ortamında paralel ve seri portlar için sorunsuz çalışmasına rağmen Windows XP ortamında seri port için çalışmamaktadır. Bu yüzden biz uygulamalarımızda API fonksiyonları ile beraber kullanacağız. Neden o zaman MsComm bileşenini ya da API fonksiyonlarını doğrudan kullanmıyoruz? Çünkü seri port yazmaçlarını daha kolay anlayabilmek ve anlatabilmek için düşük seviyeli programlamaya ihtiyacımız var. Bahsedilen düşük seviyeliden amaç, port üzerindeki her bir iğneye doğrudan erişerek esas görevinden bağımsız kullanmaktır. Örneğin, TxD hattı karakter iletimi için kullanılmasına rağmen biz onu DTR yada RTS hattı gibi sinyal kaynağı olarak kullanabiliriz.

1.7. API Fonksiyonları ile Seri Porta Erişim

Tablo 1.4'te bu modül içinde kullanılacak seri haberleşme fonksiyonları görülmektedir.

API Fonksiyonları	Açıklama
CreateFile	Seri Portu kullanıma açar.
CloseHandle	Seri Portu kapatır.
GetCommState	Seri Port özelliklerini okur.
EscapeCommFunction	Seri Porta iğnelerini yönlendirir.
BuildCommDCB	Seri Port ayarlarını yapar.
ReadFile	Seri port ya da bir dosyadan veri okur.
WriteFile	Seri porta ya da bir dosyaya veri yazar.

Tablo 1.4: API fonksiyonları

Burada sadece iki API kullanılacaktır.

İlk API, CreateFile fonksiyonudur. Windows, portları da dosyalar gibi görüyor. Aynen bir dosyayı ilk kullanıma açarken(oluştururken) neyi düşünüyorsa port için de aynı şeyi düşünüyor. Yani porttu bir dosya gibi açıp kapatıyor. Bu fonksiyonun bildirimi:

```
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
```

Programlarda kullanılacak yapı ise aşağıda görüldüğü gibidir.

```
hCom = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
```

İlk parametre, kullanılacak seri port. İkinci parametre, Windows'a portun yazma okuma amaçlı kullanılacağını söyler. Beşinci parametre port fiziksel olarak mevcut açmasını rica eder. Burada kullanılmayan tüm parametrelerin sıfır olduğuna dikkat edilmelidir. Fonksiyonun icrası sonucunda Windows, açtığı porta bir numara verir ve bu bir değişkene atar (burada hCom değişkeni). Bu numaraya handle, tutamaç ya da tutam numarası denir.

Açılan portun muhakkak kapatılması gerekir. Bunun için CloseHandle fonksiyonu kullanılır. Bildirimi:

```
Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
```

CreateFile ile elde edilen tutamaç numarasını kullanır. Aşağıda öğrenilen konuların kullanıldığı bir örnek görülmektedir.

Örnek 1.1: Seri portu açma kapama uygulaması

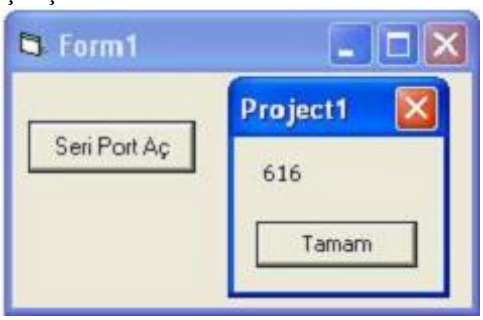
1.ADIM: Yeni bir proje başlatılır ve projeye bir modül eklenir. Modülün içine CreateFile ve CloseHandle fonksiyonlarının ve sabitlerinin bildirimi yapılır.

```
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFil
As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long,
NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long,
dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long

Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3
Public Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long)
As Long
```

2. ADIM: Form üzerine bir düğme yerleştirilir ve tıklama olayına ilişkin kodlar yazılır.

```
Private Sub Command1_Click()
Dim Ine_Bey As Long        '25 Eylül 1396
Ine_Bey = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, 0, _
 0, OPEN_EXISTING, 0, 0)
MsgBox Ine_Bey
CloseHandle Ine_Bey
End Sub
```

3. ADIM: Program çalıştırılır.

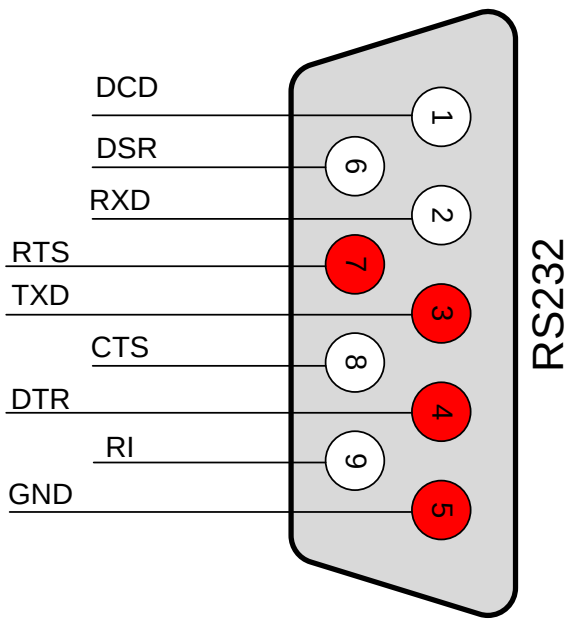


Resim 1.3: Programın çalışması

Görüldüğü gibi Windows, açılan seri porta bir numara verdi. Sizin bilgisaya-rınızda bu numara farklı olacaktır. VB'in her çalıştırılmasında bu numaranın farklı olduğu görülür.

Bu bilgiler ışığında, DTR, RTS ve TxD hatlarının gerilimi, aç-kapa tarzında değiştirilecektir. Bu amaçla Şekil 1.6'da görüldüğü gibi dişi bir seri port soketi üzerinde RTS, DTR, TxD ve GND hatlarına birer kablo lehimleyerek uzatılır. Bir ölçü aleti ile RTS, DTR ve TxD hatlarının GND hattına göre gerilim değeri ölçülerek, ölçülen değer not edilir.

ÖRNEK 1.2: DTR, RTS ve TxD uçlarının gerilim değerlerinin değiştirilmesi.

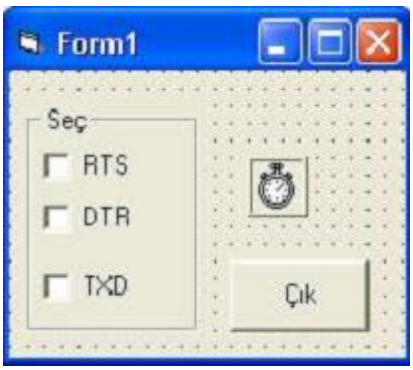


Resim 1.4: Seri porttan çıkış alınması

1. ADIM: Projeye bir modül ekleyerek API ve sabit tanımlamaları yapılır.

```
Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32" _
(ByVal PortAddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)

Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName
As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal
NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal
dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long

Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3

Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
```

2. ADIM: Form tasarımını Resim 1,5'te görüldüğü gibi yapılır ve Timer bileşeninin Interval özelliği 100-200 arasında bir değere eşitlenir.



Resim 1.5: Form tasarımı

3. ADIM: Formun "declarations" bölümüne sabit ve değişkenler tanımlanır.

```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64
Dim Nigbolu As Long
Dim yaz As String
Dim x As Long
```

4. ADIM: Formun Load olayı

```
Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_REA
ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)

yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub
```

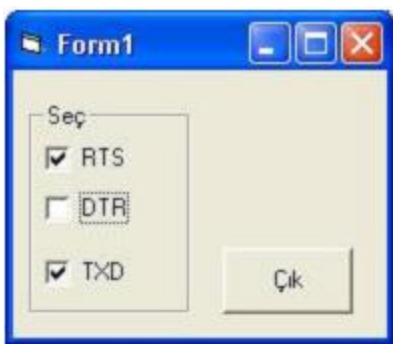
5. ADIM: Formun çıkış kodları

```
Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
 End
End Sub

Private Sub Form_Unload(Cancel As Integer)
  x = CloseHandle(Nigbolu)
End Sub
```

6 ADIM: Timer olayı.

```
Private Sub Timer1_Timer()
 If Check1.Value = 1 Then
   Out COM1 + 4, RTS
   Else
   Out COM1 + 4, 0

  End If
  If Check2.Value = 1 Then
   Out COM1 + 4, DTR
   Else
   Out COM1 + 4, 0
   End If

 If Check3.Value = 1 Then
   Out COM1 + 3, TxD
   Else
   Out COM1 + 3, 0
  End If
End Sub
```

7. ADIM: Program çalıştırılır. Seri port açıldıktan sonra onay kutuları tıklanarak ölçü aleti ile ilgili hatların gerilim değerleri ölçülür.



Resim 1.6: Programın çalışması

Görüldüğü gibi –12V seviyesinde olan hatlardaki gerilim, onay kutuları tıklandıktan sonra +12V seviyesine yükseldi.

Peki seri portun taban adresinin 3F8 olduğu söylendikten sonra neden programda taban adrese katsayı eklendi?

Seri port 8 ayrı adres alanını paylaşan 12 yazmaçtan meydana gelme-ktedir. Tablo 1.5'te bu yazmaçlar görülmektedir.

Her bir yazmaç seri port taban adresine bir katsayının eklenmesiyle bulunacak adreste bulunur. Yukarıda bahsedilen RTS ve DTR, "Modem Control Register (MCR)" içindedir ve katsayısı dörttür. Bu yüzden yazmaç adresi (H3F8+4)=H3FC olmalıdır.

TxD ucu, "Line Control Register"(LCR) isimli yazmaç üzerindedir ve kat sayısı üçtür. Bu yüzden yazmaç adresi (H3F8+3)=H3FB olmalıdır.

Yazmaç İsmi	Katsayı	Çalışma Modu
Transmitter Holding Buffer (THR)	0	Yazma
Receiver Buffer		Okuma
Divisor Latch Low Byte (DLL)		Okuma/Yazma
Interrupt Enable Register (IER)	1	Okuma/Yazma
Divisor Latch High Byte (DLM)		Okuma/Yazma
Interrupt Identification Register (IIR)	2	Okuma
FIFO Control Register		Yazma
Line Control Register (LCR)	3	Okuma/Yazma
Modem Control Register (MCR)	4	Okuma/Yazma
Line Status Register (LSR)	5	Okuma
Modem Status Register (MSR)	6	Okuma
Scratch Register	7	Okuma/Yazma

Tablo 1.5: Seri port yazmaçları

Burada da veri göndermede kullanılan TxD ucu yükseğe alçağa çekilebilen bir uç gibi kullanılmaktadır. Bu uç UART'ın LCR(Line Control Register) yazmacında yer almaktadır. Peki bu katsayılar nereden gelmektedir? Aşağıda sorunun cevabı için bu yazmaçların bit düzeni verilmiştir.

Yazmaç Adı	Ek Kat.	Bit 7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Modem Control Register	4							RTS	RTS
Line Control Register	3		TxD						

Tablo 1.6: MCR ve LCR yazmacı

Görüldüğü gibi DTR ve RTS ilk iki biti, TxD ise 6. biti işgal etmektedir. Tablo 1.7'de bu uçlara verilecek değere göre seri porta yazılacak değerler görülmeli-dir.

Yazım Vaziye	Yazım Vaziye
OUT (COM1 + 4), 1	DTR = 0, RTS = 1
OUT (COM1 + 4), 3	DTR = 0, RTS = 0
OUT (COM1 + 3), 64	TxD = 0
OUT (COM1 + 4), 1	DTR = 0, RTS = 1
OUT (COM1 + 4), 3	DTR = 0, RTS = 0
OUT (COM1 + 3), 64	TxD = 0

Tablo 1.7: RTS, DTR ve TxD kat sayıları

Yalnız yukarıdaki örnekte DTR etkinden RTS etkin olamıyor. Birini etkin kıldığın zaman diğeri etkinlikten çıkıyor. Birbirini etkilemeden istediğimiz ucu devreden çıkarıp istediğimizi etkin kılmak istersek ne yapılacaktır?

Örnek 1.3: DTR, RTS ve TxD uçlarının gerilim değerlerinin değiştirilmesi

RTS, DTR ve TxD onay kutularına sırasıyla chkRTS, chkDTR, chkTxD isimleri atanır.

Program kodları tekrar verilirse;

```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64

Dim Nigbolu As Long
Dim yaz As String
Dim x As Long
Dim y As Integer

Function Comm_set(Dogan_Bey As Integer) As Integer
  If (Dogan_Bey And TxD) = TxD Then
    Call Out(COM1 + 3, TxD)
  End If
   If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
    Call Out(COM1 + 4, (Inp(COM1 + 4) Or Dogan_Bey))
  End If
End Function
```

```
Function Comm_reset(Dogan_Bey As Integer) As Integer
  If (Dogan_Bey And TxD) = TxD Then
   'TD resetlenir
   Call Out(COM1 + 3, 0)
 End If
  If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
   'DTR ve/ya da RTS resetlenir
   Call Out(COM1 + 4, (Inp(COM1 + 4) And (Not Dogan_Bey)))
 End If
End Function

Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
End Sub

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0,
ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
  x = CloseHandle(Nigbolu)
End Sub

Private Sub Timer1_Timer()
If chkDTR.Value = 1 Then
   y = Comm_set(DTR)
  Else
  y = Comm_reset(DTR)
  End If

If chkRTS.Value = 1 Then
   y = Comm_set(RTS)
  Else
  y = Comm_reset(RTS)
```

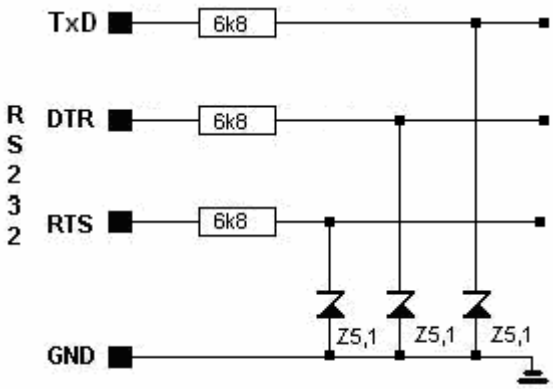
```
  End If

If chkTxD.Value = 1 Then
   y = Comm_set(TxD)
  Else
  y = Comm_reset(TxD)
End If
 End Sub
```

Burada Comm_set ve Comm_reset isimli iki fonksiyon tanımlanmıştır. Comm_set ve Comm_reset fonksiyonlarına DTR sabitinin aktarıldığı kabul edilmiştir. Aşağıda fonksiyonlara parametre girişi, sadece ilgili kısımları yazarak gözden geçirilecektir.

```
Function Comm_set(DTR) As Integer
  If (DTR And DTR) = DTR Or Then
   Call Out(COM1 + 4, (Inp(COM1 + 4) Or DTR))
 End If
End Function

Function Comm_reset(DTR) As Integer
    If (DTR And DTR) = DTR Then
   Call Out(COM1 + 4, (Inp(COM1 + 4) And (Not DTR)))
 End If
End Function
```

Birinci fonksiyonda DTR sayısı(burada 1), yazmacın o anki durumu ile OR işlemine tabi tutulmakta; ikincisinin de sayının değili(not) ile AND işlemine tabi tutulmaktadır.

Seri port çıkışlarını 5 volta düşürerek çeşitli uygulamalarda kullanmak için Şekil 1.5'te görüldüğü gibi zener diyot kullanılabilir.

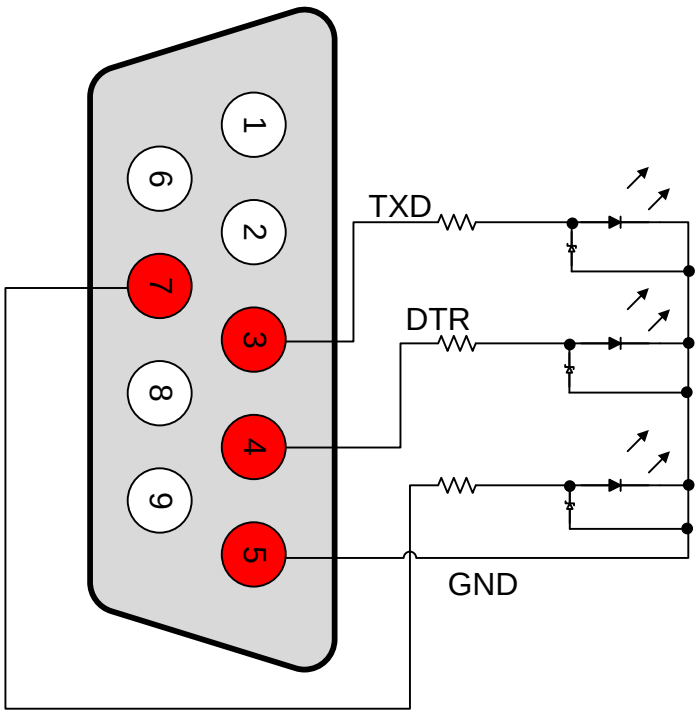


Şekil 1.5: Çıkışın 5 volta ayarlanması

Başka bir çözüm yolu olarak MAX232 yongası kullanılarak 12 volt 5 volta, 5 volt 12 volta çevrilerek bir arabirim oluşturulur.

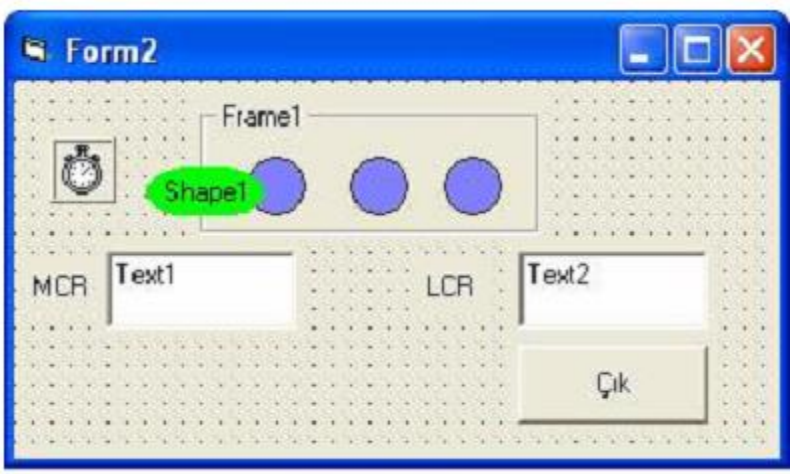
Örnek 1.4: Çıkış uçlarına LED bağlama

Şekil 1.6'da görülen devreye biraz görsellik katmak için bu uçlara LED bağlanmıştır. Devre tasarımı Şekil 1.10'da görülmektedir. Devrenin toprak hattı seri portun 5 no'lu ucuna bağlanmıştır. Bu devreyi bir breadbord üzerine ya da delikli pertenaks üzerine kurun.



Şekil 1.6: Port uçlarına led bağlama

1.ADIM: Resim 1.7'de görülen form tasarlanır.



Resim 1.7: Form tasarımı

2.ADIM: Aşağıdaki modül kısmı kodlarını yazın.

```
Public Declare Function Inp Lib "inpout32.dll" _

Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName
As
String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal
NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal
dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3

Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
```

3.ADIM: Program kodu toplu olarak aşağıda görülmektedir. Bir önceki örneğe göre Comm-set ve Comm_reset fonksiyonlarında ufak değişiklikler vardır.

```
'Yazmaç ilave katsayıları
Const Port = &H3F8
Const MCR = 4
Const LCR = 3

Const DTR = 1
Const RTS = 2
Const TxD = 64

Dim Ti As Integer
Dim PortTipi As String
Dim yaz As String
Dim Com_ID As Integer
Dim nCid As Integer
Dim x As Integer
```

```
Function Port_Kapat(nCid As Integer) As Boolean
On Error GoTo Err_Port_Kapat
Dim x As Integer  'geri dönecek değer için
Dim Netice As Boolean 'Port_Kapat için değişken

  x = CloseHandle(nCid)
  If x = 0 Then Error 1

  Netice = True
Exit_Port_Kapat:
  Port_Kapat = Netice
  Exit Function

Err_Port_Kapat:
  MsgBox "Port_Kapat fonksiyonunun icrasında hata oluştu: " & Error$
  Netice = False
  Resume Exit_Port_Kapat

End Function
Function Comm_set(Port As Integer, Alinan_Deger As Integer) As Integer
On Error GoTo Hata_Comm_set
Dim Netice As Integer

  If (Alinan_Deger And TxD) = TxD Then
    'TD
    Call Out(Port + 3, TxD)
  End If

  If (Alinan_Deger And DTR) = DTR Or (Alinan_Deger And RTS) = RTS Then
    'DTR ve/veya RTS
    Call Out(Port + MCR, (Inp(Port + MCR) Or Alinan_Deger))
  End If

  Netice = True
Terki_Comm_set:
  Comm_set = Netice
  Exit Function

Hata_Comm_set:
```

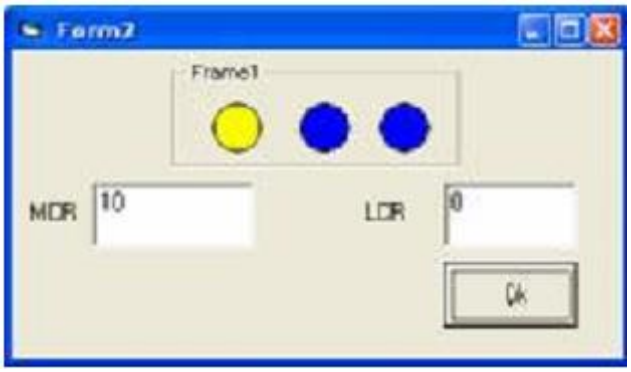
```
  Netice = False
  MsgBox "COM'DA HATA: " & Error$, 16, "Com ayarında hata"
  Resume Terki_Comm_set
End Function

Function Comm_reset(Port As Integer, Alinan_Deger As Integer) As Integer
On Error GoTo Hata_Comm_reset
Dim Netice As Integer

  Netice = False

  If (Alinan_Deger And TxD) = TxD Then
    'TD reset
    Call Out(Port + 3, 0)
  End If

  If (Alinan_Deger And DTR) = DTR Or (Alinan_Deger And RTS) = RTS Then
    'DTR ve/yada RTS resetlenir
    Call Out(Port + MCR, (Inp(Port + MCR) And (Not Alinan_Deger)))
  End If

  Netice = True

Exit_Comm_reset:
  Comm_reset = Netice
  Exit Function

Hata_Comm_reset:
  Netice = False
  MsgBox "Port Açmada Hata olustu: " & Error$, 16, "Com_Reset te hata"
  Resume Exit_Comm_reset
 End Function

Private Sub Form_Load()
Ti = 0
PortTipi = "COM1:"
  Com_ID = Port_Ac(PortTipi)
  yaz = "Açılan seri Port Numarası:" & Com_ID & vbCrLf
  MsgBox yaz
```

```
  'Port açılmadığı takdirde alt yordamı terket
  If Com_ID < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
Dim x As Integer

   x = Port_Kapat(Com_ID)
End Sub

Private Sub Terket_Click()
Unload Me
End Sub

Private Sub Timer1_Timer()
Ti = Ti + 1

If Ti = 1 Then Kirmizi
If Ti = 25 Then KirmiziTuruncu
If Ti = 50 Then Yeşil
If Ti = 75 Then Turuncu
If Ti = 100 Then Ti = 0

Text1.Text = (Inp(Port + 4))
Text2.Text = (Inp(Port + 3))
End Sub

Function Port_Ac(ByVal PortTipi As String) As Integer
On Error GoTo Err_Port_Ac
 nCid = CreateFile(PortTipi, GENERIC_READ Or GENERIC_WRITE, 0, _
0, OPEN_EXISTING, 0, 0)
  If nCid < 0 Then Error 1

Exit_Port_Ac:
  Port_Ac = nCid
  Exit Function

Err_Port_Ac:
  MsgBox "Port_Ac fonksiyonunu
```

```
 nCid = -1
 Resume Exit_Port_Ac

End Function

Sub Kirmizi()
 x = Comm_set(Port, RTS)
 Shape1.FillColor = QBColor(14)
 x = Comm_reset(Port, DTR)
 Shape2.FillColor = QBColor(9)
 x = Comm_reset(Port, TxD)
 Shape3.FillColor = QBColor(9)
End Sub

Sub KirmiziTuruncu()
 x = Comm_set(Port, RTS)
 Shape1.FillColor = QBColor(14)
 x = Comm_set(Port, DTR)
 Shape2.FillColor = QBColor(2)
 x = Comm_reset(Port, TxD)
 Shape3.FillColor = QBColor(9)
End Sub

Sub Turuncu()
 x = Comm_reset(Port, RTS)
 Shape1.FillColor = QBColor(9)
 x = Comm_set(Port, DTR)
 Shape2.FillColor = QBColor(2)
x = Comm_reset(Port, TxD)
Shape3.FillColor = QBColor(9)
End Sub

Sub Yeşil()
 x = Comm_reset(Port, RTS)
 Shape1.FillColor = QBColor(9)
 x = Comm_reset(Port, DTR)
 Shape2.FillColor = QBColor(9)
 x = Comm_set(Port, TxD)
```

```
Shape3.FillColor = QBColor(14)
End Sub
```

4.ADIM: Program çalıştırılır ve test edilir.



Resim 1.8: Program çıkışı

Programın çalışma anında MCR yazmacının içeriğinin 8, 9, 10, 11 arasında değişmektedir. Halbuki Tablo 1.5'te tekrar incelendiğinde DTR ve RTS hatlarının mantıksal değerlerinin 1 ve 2 olduğu görülür. Her ikisi de etkinken toplam değerleri 3 olur. O zaman fazlalık olan 8 değeri nereden gelmektedir?

MCR yazmacının 4 Nu.lu bacağı olan "Aux. Output 2", dâhilî olarak mantık 1 seviyesine yükseltilmektedir ama bunun uygulamalara bir etkisi yoktur.

1.8. Seri Porttan Karakter İletimi

Seri portun üç nolu TxD ucunun asıl görevi karakter iletimidir.

Bu uçtan karakter göndermenin en basit yolu seri portun taban adresine karakter doğrudan gönderilir. Bu dll dosyasının içinde tanımlanan OUT fonksiyonunu kullanarak, WriteFile API fonksiyonu ile ya da MsComm bileşeni ile olur.

Örnek 4'te kullanılan devre, seri porta bağlı olsun. Form üzerine bir metin kutusu ve bir düğme yerleştirilir ve aşağıdaki kodlar yazılır. Modül kısmı yukarıdaki örnekle aynı olacaktır.

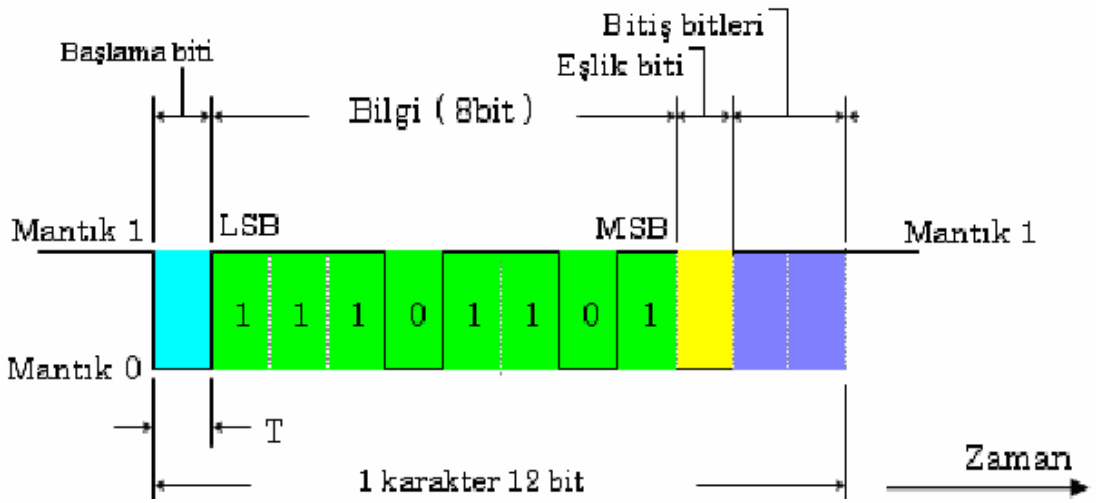
```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64

Dim Varna As Long

Dim x As Long

Private Sub Command1_Click()
 x = CloseHandle(Varna)
End Sub

Private Sub Form_Load()
Varna = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, _
 ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
If Varna < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
  x = CloseHandle(Varna)
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
Char = Chr(KeyAscii)
  KeyAscii = Asc(UCase(Char))
Out COM1, KeyAscii
End Sub
```

Program çalıştırıldığında metin kutusuna yazılan her karakter önce büyük harfe çevrilecek, arkasından seri porta gönderilecektir. Bir şeylerin porta gittiği, ledin çok kısa süre yanıp söndüğünden anlaşılır fakat karakter iletimi biraz daha yavaşlatabilir mi?

Seri portun haberleşme anında kullandığı karakter iletim süreci yakından incelenmek istenirse;

Haberleşmenin olmadığı durumlarda gönderici hat (TxD), mantık 1 seviyesindedir. (-12V) Bir bakıma hat beklemededir. Haberleşmenin başlangıcın-da bir bitin iletim süresi kadar bu hat mantık 0 seviyesine çekilir (Şekil 1.7). Bu ilk bite başlangıç biti denir. Bu aynı zamanda karşı tarafa bilgi gönderileceğini göstermektedir. Arkasından anlaşmayla belirlenen bit sayısı (5 bit, 7 bit, 8 bit) kadar bit gönderilir. 7 bit seçilirse ancak 127'ye kadar olan ASCII karakterler gönderilebilir.

Bunu eğer belirtilmişse bir eşlik (parity) biti takip eder.



Şekil 1.7: Bit düzeni

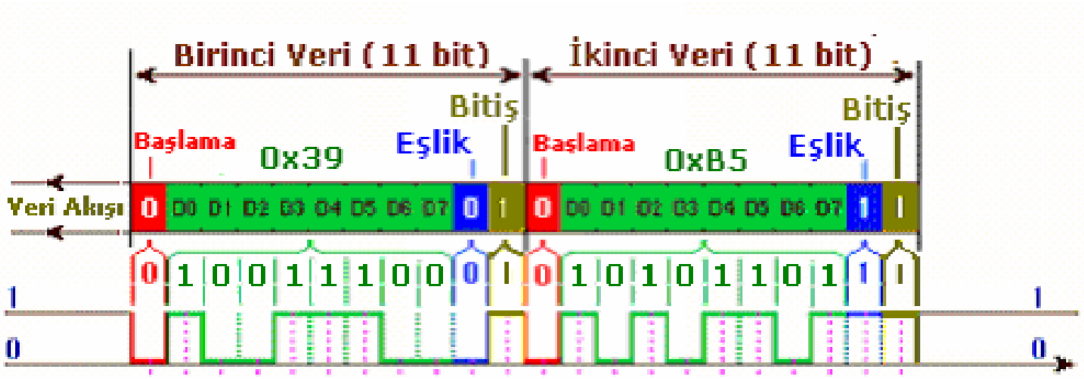
Eşlik biti, karakterin düzgün aktarılıp aktarılmadığını kontrol etmek için kullanılır. Veri paketinde 8. bittir. Eşlik biti "tek eşlik biti", "çift eşlik biti" ya da "Eşlik biti yok." değerlerini alabilir. Eşlik biti "ÇİFT" seçilirse kendisinden önce gönderilen veri bitlerinin tek ya da çift olmasına göre 0 veya 1 değerini alır. Örneğin "0110 0011" düzeninde "1" olan bitlerin sayısı çift sayıdır. Dolayısıyla bu çift sayı düzenini bozmamak için eşlik bitinin değeri "0" olur. Bit düzeni "0100 0011" ise çifte tamamlamak için eşlik bitinin değeri "1" olacaktır.

Gönderilen verilerin ikilik karşılıkları ve eşlik bitleri aşağıdaki Tablo 1.8'de görülmektedir.

Sayı	Veri Biti	Çift Eşlik	Tek Eşlik Biti
&H39	00111001	0	1
&HB5	10110101	1	0
&H1A	00011010	1	0
&H0	00000000	0	1

Tablo 1.8:Eşlik biti

Sonuç olarak gönderici hat mantık 1 seviyesine çekilerek yine anlaşmaya bağlı olarak 1 veya 2 durdurma biti gönderilir. Aktarımı yapılacak diğer karak-ter için aynı işlem takip edilir. Şekil 1.8'de iki bitin ardışık gönderimi temsil edil-miştir.



Şekil 1.8: Ardışık bitlerin gönderilmesi

Buna göre bir karakterin iletimi için seçime bağlı olarak 10, 11 veya 12 bit gerekiyor. Peki bu bitler hangi hızda iletilecek ve alıcı hangi hızda alacak? Bunun alıcı ve verici tarafından bilinmesi gerekir. Bu hıza **baud hızı** denir.

Baud hızı, ismini, 1890'lı yıllarda Fransız Telgraf İşletmesi'nde çalışmakta olan Jean Maurice Emile Baudot'tan almıştır. Bir hattın saniyedeki durumunu değiştirme oranı olarak tarif edilen baud hızı, günlük hayatta saniyede iletilecek bit sayısı olarak bilinir. Baud hızı 19200 ise saniyede 19200 bit iletilecek ya da 19200 defa hat seviye değiştirecek anlamındadır. Baud hızı ile seri haberleşme mesafesi Tablo 1.9'da görülmektedir.

Baud Hızı	Yalıtılmış Kablo [m]	Yalıtılmamış Kablo [m]
110	5000	1000
300	4000	1000
1200	3000	500
2400	2000	500
4800	500	250
9600	250	100

Tablo 1.9: Baud hızına bağlı iletim mesafesi

Hem kullanıcı hem de alıcı baud hızı ile beraber bu veri biti düzeninde de birbiriyle anlaşmış olması gerekir. Aksi takdirde veriler anlamsız ve karmakarışık alınır.

Bilgisayarlarda 1.8432 MHz'lik saat osilatörü bir ön bölücü ile 16'ya bölünerek 115200 Hz elde edilir. Bu değer program yoluyla bölünerek istenilen baud hızı elde edilir. Bölen sayı iki bayt hâlinde DLM ve DLL yazmaçlarında saklanır. DLM, büyük baytı saklar. Tablo 1.10, bu sayıları göstermektedir.

Baud Hızı	Bölen Sayı	DLM	DLL
110	1047	04H	17H
300	384	01H	80H
600	192	00H	C0H
1200	96	00H	60H
2400	48	00H	30H
3600	32	00H	20H
9600	12	00H	0CH
19200	6	00H	06H
38400	3	00H	03H
57600	2	00H	02H

Tablo 1.10: Baud hızı

Baud hızını DLL ve DLM yazmacına yazabilmek için LCR yazmacının son biti, Tablo1.11'de görüldüğü gibi "1" yapılmalıdır. Bu da OUT (COM1 + 3 , 0x80) komutu ileyapılmaktadır. Daha sonra bu yazmacın alt bitlerine veri biti sayısı, eşlik biti gibi değerler yerleştirilmektedir.

Formun Load olayı aşağıdaki gibi değiştirilerek program çalıştırılır.

```
Private Sub Form_Load()
Varna = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, _
 ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
If Varna < 0 Then Exit Sub

Out (COM1 + 1), 0
Out (COM1 + 3), &H80
Out (COM1 + 0), &H17
Out (COM1 + 1), &H4   '110 baud hızı
Out (COM1 + 3), &H3 '8 veri biti, eşlik biti yok, 1 stop biti
End Sub
```

Bir örnek vermek gerekirse DLM yazmacına 00, DLL yazmacına 0C yazıldığında 115200 Hz frekans 12'ye bölünerek 9600 Hz baud hızı elde edilir. Bu da saniyede 9600 ardışık bit iletilebilir anlamına gelmektedir. Bir bitlik süre 1 / 9600 = 104.2 mikro saniyedir. 8 veri biti, 1 başlangıç, 1 sonlandırma bitinden oluşan gönderilecek bir karakter 10 bit uzunluğundadır ve bu bilgi 10 * 104.2 = 1042 mikro saniyede gönderilir. Bu da yaklaşık olarak 1 mili saniye yapar.

Yukarıdaki programın çalıştırılması sonunda karakterlerin iletimi görülmektedir. Peki UART'ın kabul edeceği en düşük baud hızı nedir? 115200/57600 =2. Buna göre bölen olan 57600 sayısının onaltılık karşılığı E100 sayısını DLM ve DLL yazmaçlarına paylaştırılacaktır.

Aşağıdaki değişiklikler yapılarak program bir kez daha çalıştırılır.

```
Out (COM1 + 1), 0
Out (COM1 + 3), &H80
Out (COM1 + 0), &H0
Out (COM1 + 1), &HE1
Out (COM1 + 3), &H3
```

Artık, basılan tuşun bitleri görülmektedir.

LCR yazmacı, haberleşme için gerekli olan parametreleri ayarlar. Bu yazmaçta verinin bit uzunluğu, eşlik biti sayısı, başlangıç ve bitiş bitleri ayarlan-maktadır. Tablo 1.11, bu bitlerin durumunu göstermektedir.

Bit 7	1	Bölme yazmacı Erişim Biti		
	0	Alıcı ve verici tamponuna, Kesme Yetki Yazmacına Erişim		
Bit 6	Fasıla Yetkisi			
Bit 5 Bit 4 Bit 3	Bit 5	Bit 4	Bit 3	Eşlik Seçimi
	X	X	0	Eşlik Yok
	0	0	1	Tek Eşlik
	0	1	1	Çift Eşlik
	1	0	1	Yüksek Eşlik
	1	1	1	Alçak Eşlik
Bit 2	Durdurma Bitinin Uzunluğu			
	0	Bir adet durdurma biti		
	1	6, 7, 8 bit uzunluğundaki kelime uzunluğu için 2 durdurma biti, 5 bit uzunluğunda kelime için 1.5 durdurma biti		
Bit 1 Bit 0	Bit 1	Bit 0	Kelime uzunluğu	
	0	0	5 bit	
	0	1	6 bit	
	1	0	7 bit	
	1	1	8 bit	

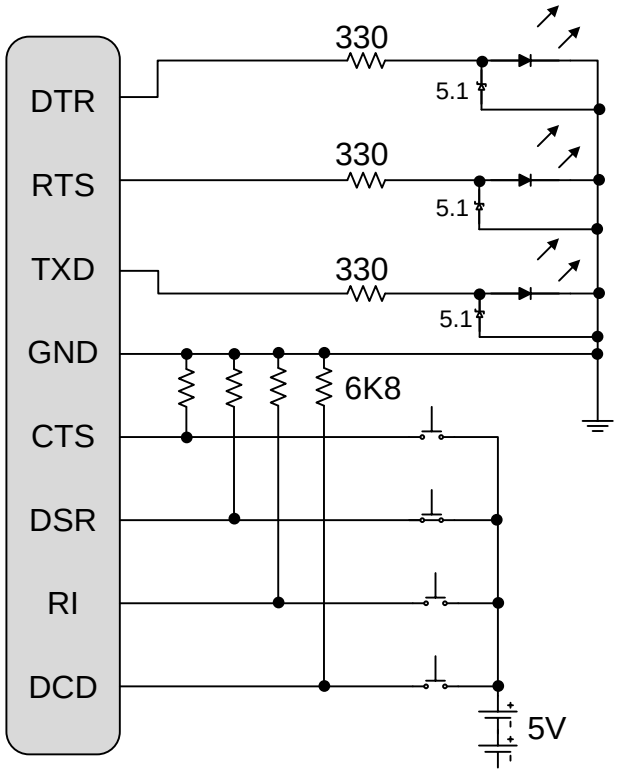
Tablo 1.11: LCR yazmacı

Karakter iletimin de yukarıda yaptığımız işlemi, BuildCommDCB, SetCommState ve GetCommState API fonksiyonları da yapmaktadır.

UYGULAMA FAALİYETİ

Aşağıdaki sorulara ilişkin uygulama faaliyetini yapınız.

- Aşağıda görülen devreyi yaparak genel amaçlı seri port giriş-çıkış devresi yapınız. Burada butonlar yerine sensörler bağlanabilir. Her bir butona basılma sayısını ekranda gösteren programı yazınız.



- DTR, RTS ve TxD hatlarına bağlı ledlerin yarımşar saniye aralıklarla onar kere yanmasını sağlayan programı yazınız.

İşlem Basamakları	Öneriler
➢ Form tasarımını yapınız. Bileşenlere uygun isimler veriniz. ➢ Programı yazınız. ➢ Yazdığınız programı derleyiniz. ➢ Programda hata var ise bunları düzeltiniz. ➢ Ekran görüntüsünü kontrol ediniz.	➢ Programda kullanacağınız. değişkenlerin tipini belirlemelisiniz. ➢ Değişken isimlendirme kurallarına dikkat etmelisiniz. ➢ Program satırlarının düzenli olmasına özen göstermelisiniz. ➢ Karar ifadelerinin belirlenen şartlara uygun olmasına dikkat etmelisiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. RS-232 standardı hangi portlar için geliştirilmiştir?
A) Paralel portlar
B) USB portlar
C) Seri portlar
D) EPP portları
E) Paralel portlar

2. **2.** Aşağıdakilerden hangisi seri port çıkış ucudur?
A) DTR
B) DSR
C) DCD
D) CTS
E) GND

3. Seri portta Mantık 1'in gerilim değeri kaç volttur?
A) -12V
B) 12V
C) -3V
D) 5V
E) -5V

4. COM1'in taban adresi kaçtır?
A) 2F8
B) 3F8
C) 2F9
D) 378
E) HH6

5. DLAB biti hangi yazmaç üzerindedir?
A) MSR
B) MCR
C) LSR
D) LCR
E) LRC

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ–2

AMAÇ

Seri port donanımı yoluyla sayısal giriş değerlerini kontrol edebileceksiniz.

ARAŞTIRMA

- Seri portla cihazların kontrolünü araştırınız.
- Seri port uygulama komutlarını araştırınız.
- Seri port için yazılmış ugulamaları araştırınız.

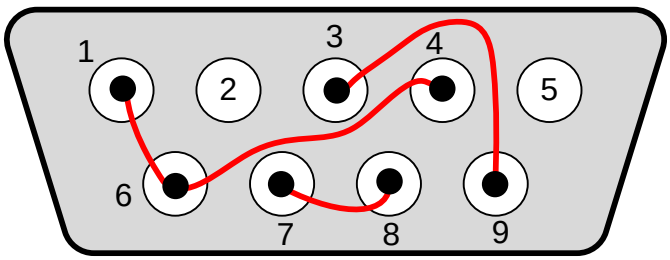
2. SERİ PORTA GİRİŞ YAPILMASI

Bir önceki bölümde seri port uçlarından nasıl çıkış yapıldığını ve nasıl karakter iletimi yapıldığını gördük. Seri port yolundaki yolculuğumuz devam etmek-tedir.

2.1. Seri Porttan Sinyal Okuma

Seri porta giriş için TTL mantık seviyesi doğru olarak algılandığından giriş gerilimini 12 volta yükseltmeye gerek yoktur. Doğrudan giriş yapılabilir.

Burada seri portun giriş uçlarını kullanmak için seri portun çıkış uçlarından yararlanacağız. Bu bakımdan dişi bir seri port konnektörü üzerinde Şekil 2.1'de görülen bağlantılar yapılarak seri port üzerine takılır.



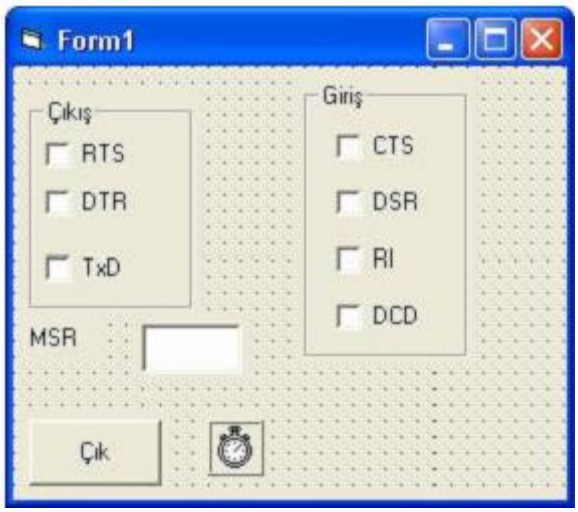
Şekil 2.1: Giriş bağlantısı

Program yazımına başlamadan önce kısa bir not: Seri portta 2 no'lu RxD ucu, genel veri okuma ucudur. Bunun dışındaki 1,6,8 ve 9 no'lu uçlar, giriş uçlarıdır. RxD ucundan farkları, sinyal seviyeleri Mantık 1 ve Mantık 0 arasında değişmesi yani aç-kapa tarzında çalışmasıdır. RxD ucunun çalışması, bir sonraki bölümde görüleceği gibi biraz farklıdır.

ÖRNEK 2.1: Seri port giriş uçlarının kullanılması.

Örnek 1.2'nin üzerine ilaveler yapılacaktır.

ADIM 1: Form tasarımı Şekil 2.1'de görüldüğü gibi yapılır.



Şekil 2.2: Form tasarımı

Bileşenlere, başlıklara uygun olarak chkRTS, chkDTR, chkTxD, chkCTS, chkDSR, chkRI, chkDCD ve txtMSR isimleri atanır. Timer'ın Interval özelliğine de 100 verilir.

ADIM 2: Programın kodlarını yazınız.

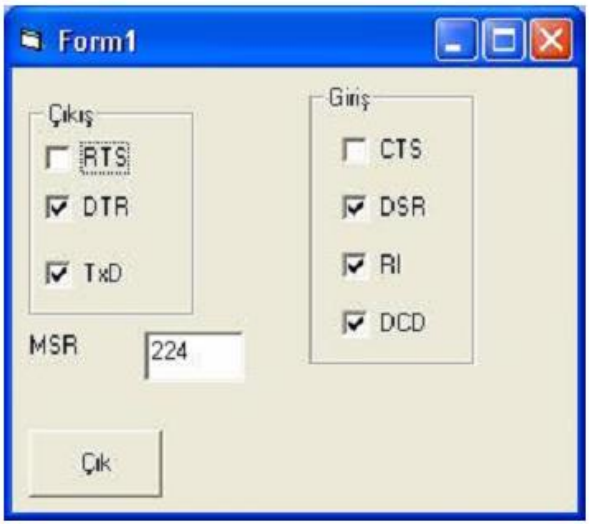
```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64

Const CTS = 16
Const DSR = 32
Const RI = 64
Const DCD = 128

Dim Nigbolu As Long
Dim yaz As String
Dim x As Long
Dim y As Integer

Function Comm_set(Dogan_Bey As Integer) As Integer

  If (Dogan_Bey And TxD) = TxD Then
```

```
   Call Out(COM1 + 3, TxD)
  End If

  If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
   Call Out(COM1 + 4, (Inp(COM1 + 4) Or Dogan_Bey))
  End If

End Function

Function Comm_reset(Dogan_Bey As Integer) As Integer

  If (Dogan_Bey And TxD) = TxD Then
   Call Out(COM1 + 3, 0)
  End If

  If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
   'DTR ve/veya RTS resetlenir
   Call Out(COM1 + 4, (Inp(COM1 + 4) And (Not Dogan_Bey)))
  End If
End Function

Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
 End
End Sub

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE,
ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
  x = CloseHandle(Nigbolu)
End Sub

Private Function Comm_Ahval(Minnet_Bey As Integer) As Boolean
On Error GoTo Hatali_Durum
Comm_Ahval = ((Inp(COM1 + 6) And Minnet_Bey) = Minnet_Bey)

Comm_Ahval_Terk:
Exit Function
```

```
Hatali_Durum:
MsgBox "Port durumunu Öğrenirken Hata Oldu. Hata No:" & Error$
Resume Comm_Ahval_Terk
End Function
Private Sub Timer1_Timer()

txtMSR.Text = Inp(COM1 + 6)

If chkDTR.Value = 1 Then
   y = Comm_set(DTR)
  Else
  y = Comm_reset(DTR)
  End If

If chkRTS.Value = 1 Then
   y = Comm_set(RTS)
  Else
  y = Comm_reset(RTS)
  End If

If chkTxD.Value = 1 Then
   y = Comm_set(TxD)
  Else
  y = Comm_reset(TxD)
End If

If Comm_Ahval(CTS) = True Then
 chkCTS.Value = 1
 Else
 chkCTS.Value = 0
 End If

 If Comm_Ahval(DSR) = True Then
 chkDSR.Value = 1
 Else
 chkDSR.Value = 0
 End If

 If Comm_Ahval(RI) = True Then
 chkRI.Value = 1
 Else
 chkRI.Value = 0
 End If

 If Comm_Ahval(DCD) = True Then
```

```
chkDCD.Value = 1
Else
chkDCD.Value = 0
End If

End Sub
```

ADIM 3: Program çalıştırılır.



Şekil 2.3: Programın çalışması

Burada seri portun çıkış uçları portun giriş uçlarına bağlanarak sinyal okunmuştur. Port giriş uçları Modem Status Register(MSR) üzerindedir. Okunan değer bit düzenine göre "And" işlemine tabi tutularak değerinin bir veya sıfır olduğuna karar verilmektedir.

Yazmaç Adı	İlave Kat.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Modem Status Register	6	DCD	RI	DSR	CTS				
Bit Değerleri →		128	64	32	16				

Tablo 2.1: MSR yazmacı

Seri portun DTR, RTS ve TxD hatlarını doğrudan süren EscapeCommFunction isimli bir API tanımlanmıştır. Bu seri porta bazı nitelikler yükler. Bildirimi:

Declare Function EscapeCommFunction & Lib "kernel32" (ByVal nCid As Long, _ ByVal nFunc As Long)

Aldığı parametreler Tablo 2.2'de görülmektedir.

Parametre	Tanım
nCid	Açık olan seri port numarası.
nFunc	İcra edilmek üzere aşağıda niteliği belirtilen fonksiyonu gösteren sabit ➢ CLRBREAK: ClearCommBreak API fonksiyonu ile aynı işi yapar. TxD hattını açar. ➢ CLRDTR: DTR hattını kapatır. ➢ CLRRTS: RTS hattını kapatır. ➢ SETBREAK: SetCommBreak fonksiyonu aynı işi yapar. TxD hattını kapatır. ➢ SETDTR: DTR ucunu etkin yapar. ➢ SETRTS: RTS ucunu etkin kılar. ➢ SETXON: XOn karakteri alınmış gibi portun davranmasını sağlar. ➢ SETXON: XOff karakteri alınmış gibi portun davranmasını sağlar.

Tablo 2.2: EscapeCommfunction fonksiyonu

Yukarıdaki örnekte MSR yazmacı, adresi doğrudan yazılarak okundu. Bu yazmacı okuyan GetCommModemStatus isimli bir fonksiyon vardır. Bildirimi:

Declare Function GetCommModemStatus& Lib "kernel32" _
(ByVal hFile As Long, lpModemStat As Long)

Parametreleri tablo 2.3'te görülmektedir.

Parametre	Tanım
hFile	Açık olan seri port numarası.
lpModemStat	Modem durumunu gösteren değişken aşağıdaki sabitlerin birleşimi olabilir. ➢ MS_CTS_ON: CTS açık ➢ MS_DSR_ON: DSR açık ➢ MS_RING_ON: RI sinyali hissedildi. ➢ MS_RLSD_ON: DCD sinyali hissedildi.

Tablo 2.3: GetCommModemStatus fonksiyonu

Aşağıda bu fonksiyonlar bir örnekle desteklenmektedir.

Örnek 2.2: Seri porta girişlerini API fonksiyonları ile denetlemek

Form tasarımı, Örnek 1.1 ile aynı olacak.

ADIM 1: Modül kısmı:

```
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA"_
(ByVal lpFileName As String, ByVal dwDesiredAccess As Long, _
ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, _
ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, _
ByVal hTemplateFile As Long) As Long

Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3

Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long

Declare Function EscapeCommFunction Lib "kernel32" (ByVal nCid As Long, _
ByVal nFunc As Long) As Long

Declare Function GetCommModemStatus Lib "kernel32" _
(ByVal hFile As Long, lpModemStat As Long) As Long
```

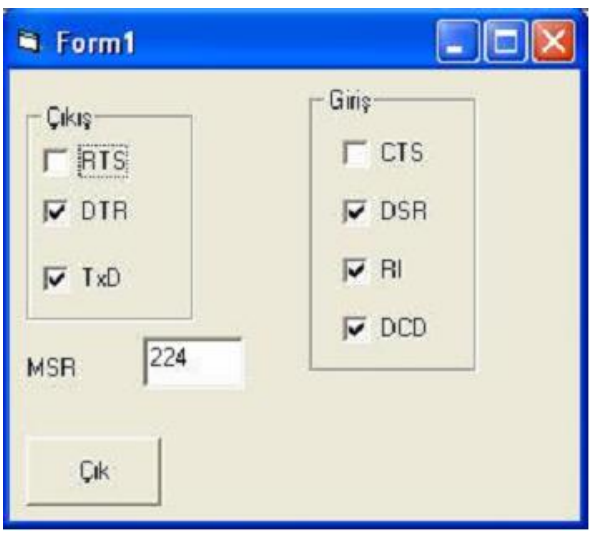
ADIM 2: Aşağıdaki kodlar yazılır.

```
Const SETRTS = 3
Const CLRRTS = 4
Const SETDTR = 5
Const CLRDTR = 6
Const SETBREAK = 8
Const CLRBREAK = 9
Const MS_CTS_ON = &H10
Const MS_DSR_ON = &H20
Const MS_RING_ON = &H40
Const MS_DCD_ON = &H80
Dim Kor_Jan As Long
Dim Minnet_Bey As Integer

Dim Nigbolu As Long
Dim yaz As String
Dim x As Long
```

```
Private Sub chkDTR_Click()

If chkDTR.Value = 1 Then
 Minnet_Bey = EscapeCommFunction(Nigbolu, SETDTR)
   Else
Minnet_Bey = EscapeCommFunction(Nigbolu, CLRDTR)
   End If
End Sub

Private Sub chkRTS_Click()
If chkRTS.Value = 1 Then
  Minnet_Bey = EscapeCommFunction(Nigbolu, SETRTS)
Else
Minnet_Bey = EscapeCommFunction(Nigbolu, CLRRTS)
   End If
End Sub

Private Sub chkTxD_Click()
If chkTxD.Value = 1 Then
Minnet_Bey = EscapeCommFunction(Nigbolu, SETBREAK)
Else
Minnet_Bey = EscapeCommFunction(Nigbolu, CLRBREAK)
End If
End Sub

Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
 End
End Sub

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0,
ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
  x = CloseHandle(Nigbolu)
End Sub

Private Sub Timer1_Timer()
Minnet_Bey = GetCommModemStatus(Nigbolu, Kor_Jan)
txtMSR.Text = Kor_Jan
```

```
If Kor_Jan And MS_CTS_ON Then
chkCTS.Value = 1
Else

chkCTS.Value = 0
End If

If Kor_Jan And MS_DSR_ON Then
chkDSR.Value = 1
Else
chkDSR.Value = 0
End If

If Kor_Jan And MS_RING_ON Then
chkRI.Value = 1
Else
chkRI.Value = 0
End If

If Kor_Jan And MS_DCD_ON Then
chkDCD.Value = 1
Else
chkDCD.Value = 0
End If
End Sub
```

ADIM 3: Programın çalışma görüntüsü

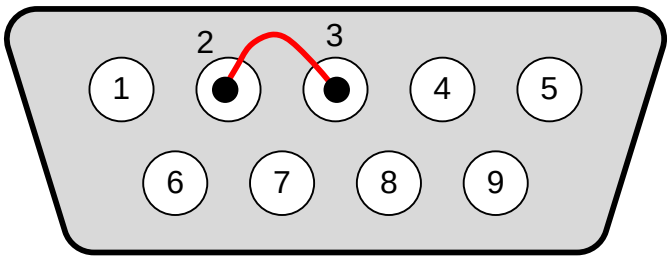


Şekil 2.4: Programın çalışması

2.2. MsComm Bileşeni

Visual Basic (VB), diğer dillerden farklı olarak seri port haberleşme için hazır bir bileşen sunar. Microsoft Comm Control olarak adlandırılan bu bile-şen seri portla ilgili bir çok ayarı beraberinde getirmektedir.

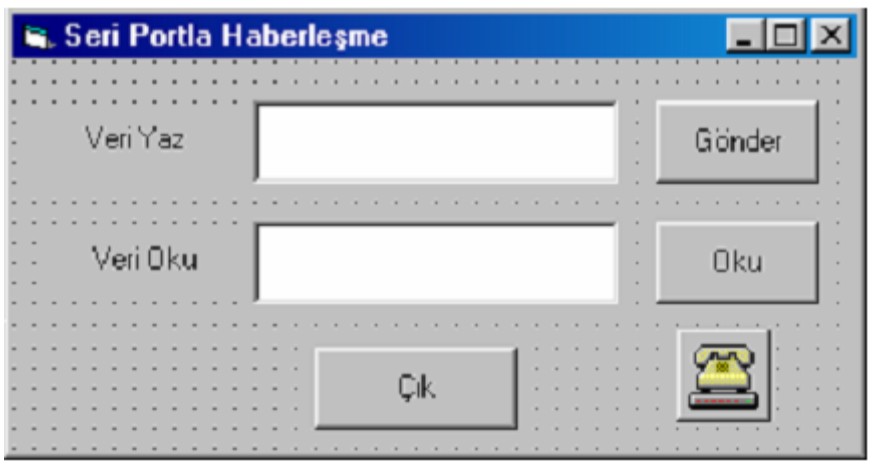
Kullanılacak her bir seri port için form üzerine bir MsComm bileşeni konur. MsComm ile ilgili teorik bilgi bir örnekle anlatılacaktır. Bu amaçla seri port soketinde TxD ve RxD uçlarına Şekil 2.4'te görüldüğü gibi küçük bir kablo lehim-lenmelidir.



Şekil 2.5: Basit bağlantı

ÖRNEK 2.3: Gönderilen karakterin geri alınması

Şekilden de anlaşılacağı gibi çıkış ucu giriş ucuna bağlanmıştır. Bu demektir ki giden her şey geri dönecektir. Bu amaçla Şekil 2.4'te görülen form tasarımını Tablo 2.4'ten yaralanarak yapınız.



Şekil 2.6: Form tasarımı

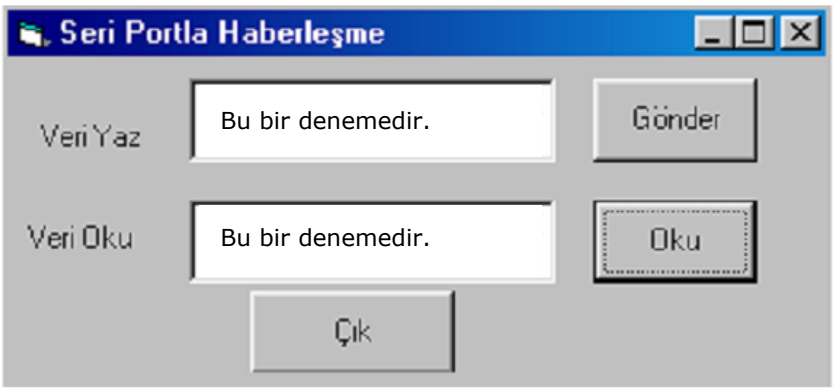
	Name	MultiLine
1. Metin Kutusu	Gonder	True
2. Metin Kutusu	Al	True
Gönder Düğmes	btnGonder	
Oku Düğmesi	btnAl	
Çık Düğmesi	btnExit	

Tablo 2.4: Form tasarım tablosu

Program kod satırları aşağıdaki gibi düzenlenir.

```
Private Sub btnGonder_Click()
MSComm1.Output = Gonder.Text + Chr$(13)
End Sub

Private Sub btnOku_Click()
Al.Text = MSComm1.Input
End Sub

Private Sub btnExit_Click()
End
End Sub

Private Sub Form_Load()
MSComm1.PortOpen = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub
```

Programı çalıştırma vakti geldi. I. Metin kutusuna bir şeyler yazılarak "Gönder" düğmesine basılır. Akabinde "Oku" düğmesi tıklanarak gönderilen verinin okunması sağlanır.



Şekil 2.7: Programın çalışması

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- Yazacağınız herhangi bir metni MsComm bileşeniyle gönderen ve gönderdiği bilgiyi alan bir program yapınız.

Burada adım motorunu sürecek olacak uçlar veri hattının 0-3 no'lu (2-5 no'lu pinler) uçlarıdır. ULN2803 Darlington bağlantılı bir sürücü olup 500 mA akım tedarik edebilmektedir. Besleme hattının toprağı ile bilgisayarın toprak hattının birleştirildiğine dikkat ediniz.

İşlem Basamakları	Öneriler
➢ Form tasarımını yapınız. ➢ Bileşenlere uygun isimler veri-niz. ➢ Programı yazınız. ➢ Yazdığınız programı derleyiniz. ➢ Programda hata var ise bunları düzeltiniz. ➢ Ekran görüntüsünü kontrol edi-niz.	➢ Programda kullanacağınız değişken-lerin tipini belirlemelisiniz. ➢ Değişken isimlendirme kurallarına dikkat etmelisiniz. ➢ Program satırlarının düzenli olmasına özen göstermelisiniz. ➢ Karar ifadelerinin belirlenen şartlara uygun olmasına dikkat etmelisiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. Aşağıdaki uçlardan hangisi Modem Status Register'ın elemanlarından değildir?
A) CTS
B) DCD
C) DSR
D) DTR
E) DRS

2. Aşağıdakilerden hangisi MsComm'un haberleşmeyi yönettiği yollardan biridir?
A) Open
B) Click
C) CommEvent
D) Load
E) Save

3. Seri porttaki genel veri okuma ucu nedir?
A) RxD
B) TxD
C) RTS
D) RI
E) DCD

4. Seri portun DTR, RTS ve TxD hatlarını doğrudan süren API'nin ismi nedir?
A) CommFunction
B) EscapeCommFunction
C) nCid
D) nFunc
E) pModemStat

5. Gönderilen karakterin geri alınması için RS232 portun hangi uçları birleştirilir?
A) 1-3
B) 4-6
C) 2-3
D) 3-4
E) 4-5

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ–3

AMAÇ

İki bilgisayar arasında veri iletişimini hatasız olarak sağlayabileceksiniz.

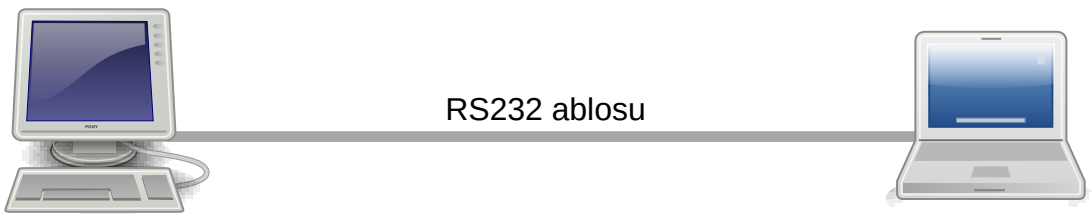
ARAŞTIRMA

- Bilgisayarlar arası dosya aktarım,
- Z modem protokolü,
- CRC hata denetimini araştırınız.

3. İKİ BİLGİSAYARI BAĞLAMAK

Bilgisayarları birbirine bağlayarak veri alışverişi yapmak en heyecan verici uygulamalardan biridir. Mekâna ve imkâna bağlı olarak bilgisayarlar birbirine çeşitli türlerde bağlanabilmektedir. Şekil 3.1 ve 3.2'de farklı iki tür görülmektedir.

Şekil 3.1: Modem Bağlantısı

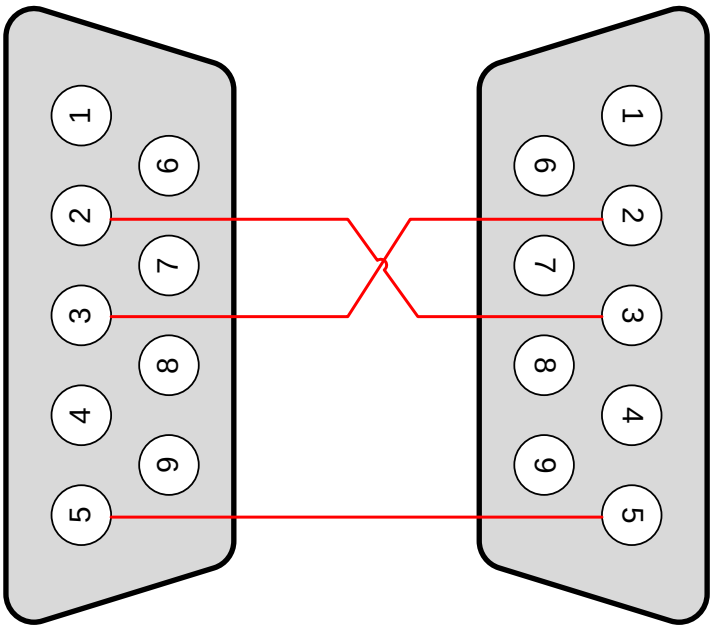


Şekil 3.2: Seri kablo bağlantısı

3.1. Seri Haberleşme Standartları

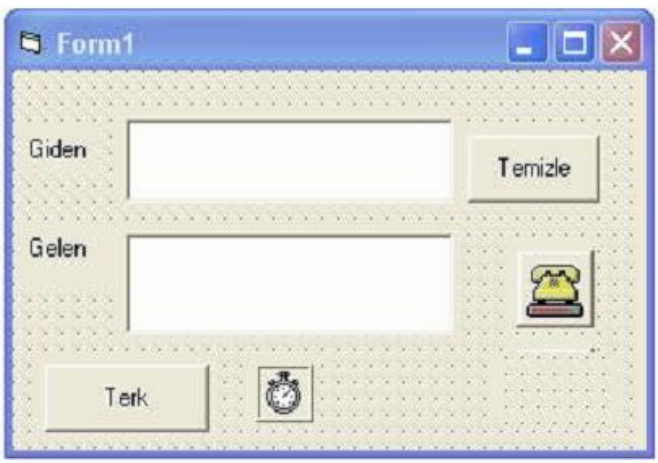
3.1.1. Bağlantı Şekilleri

Bilgisayarları seri port üzerinden bağlamanın en kısa yolu üç telli bağlama-dır. Şekil 3.3'te görüldüğü gibi TxD hattı diğer bilgisayarın RxD hattına, RxD hattı da TxD hattına bağlanır. Toprak hatları birleştirilir.



Şekil 3.3: Üç telli bağlantı

Örnek 3.1: Şekil 3.4'te görülen form tasarımı her iki bilgisayara da yapılır.

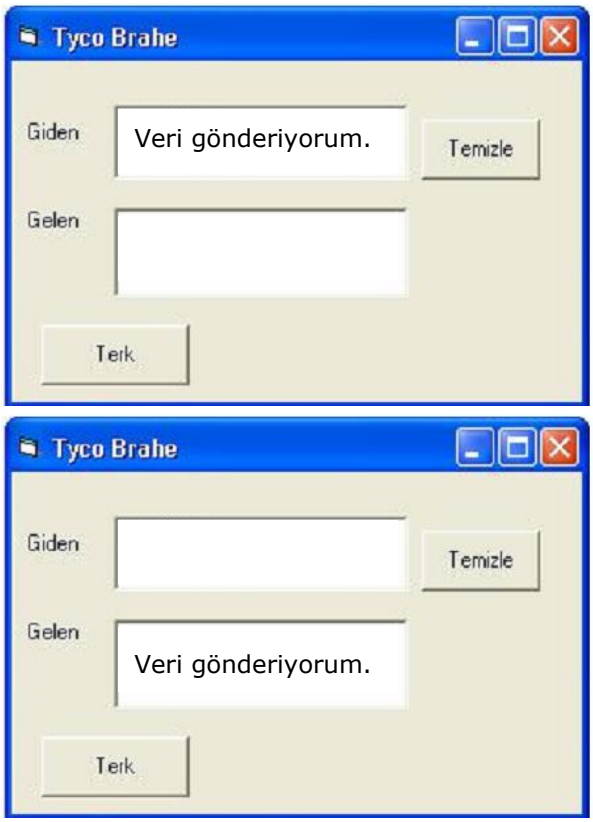


Şekil 3.4: Form tasarımı

Şekil 3.4'te görülen bağlantı yapılır ve seri portlar bilgisayarlara bağlanır.

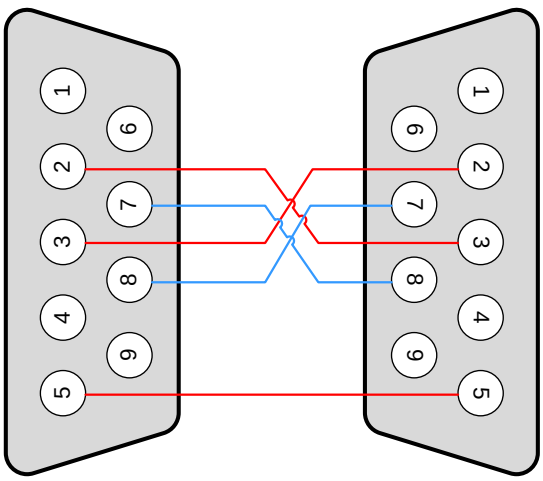
```
Private Sub Command1_Click()
Unload Me
End Sub

Private Sub Command2_Click()
Text1.Text = ""
MSComm1.Output = "Sil"
End Sub

Private Sub Form_Load()
MSComm1.PortOpen = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
MSComm1.Output = Chr$(KeyAscii)
End Sub

Private Sub Timer1_Timer()
Selam = MSComm1.Input
If Selam = "Sil" Then
Text2.Text = ""
ElseIf Selam <> "Sil" Then
Text2.Text = Text2.Text + Selam
End If
End Sub
```

Yukarıdaki kodları yazdıktan sonra program çalıştırılır.



Şekil 3.5: Programın çalışması

İki bilgisayar arasında basit bir donanım el sıkışmalı bir bağlantı kurula-caksa Şekil 3.6'da görülen kablo bağlantısı kullanılmalıdır.

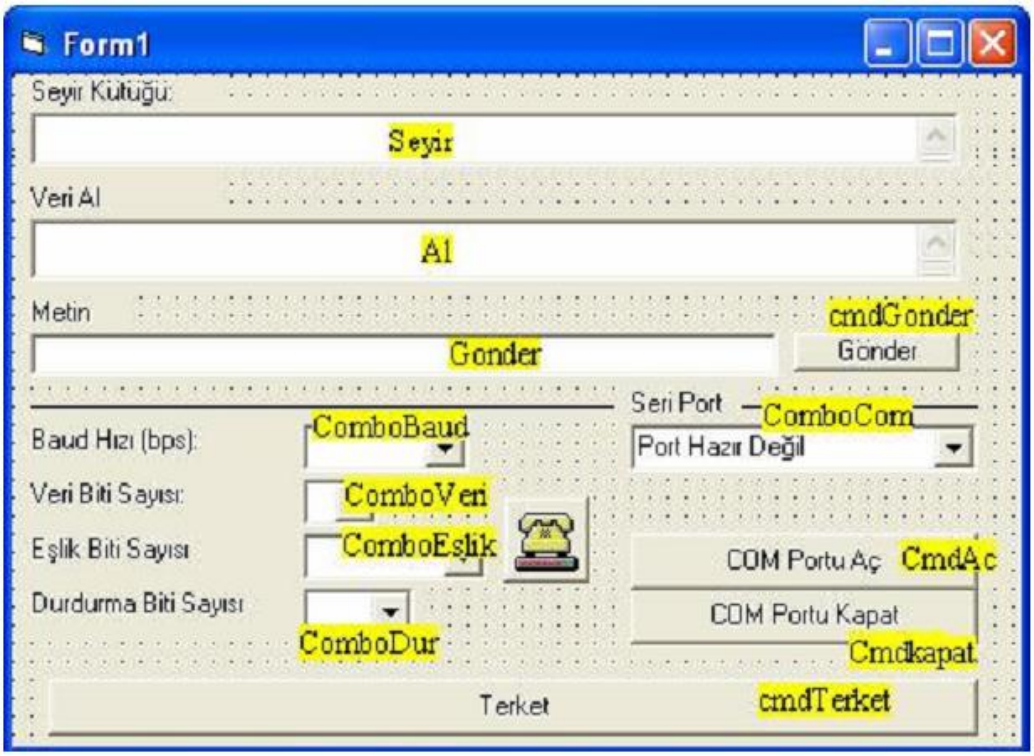


Şekil 3.6: El sıkışmalı bağlantı

Örnek 3.2: Seri portla iki bilgisayarı irtibatlandırma

Bu amaçla Şekil 3.6'da görülen bağlantıyı uzun kablolarla yaparak bilgisayarları seri portlarına takınız.

Form tasarımı Şekil 3.7'de görüldüğü gibi yapılarak gerekli isim atamaları yapılır.



Şekil 3.7: Form Tasarımı

Yukarıdaki kod satırlarını yazdıktan sonra programımızı çalıştıralım ve deney kartı üzerindeki butonlara sırayla basarak aşağıdaki tabloyu doldurunuz.

	Name	MultiLine
1. Metin Kutusu	Seyir	True
1. Metin Kutusu	Al	True

Tablo 3.1: Ayar tablosu

Formun kod bölümü:

```
Option Explicit
Dim PortOpen As Boolean

Public Function PortTest(COMPortNummer As Integer) As Boolean
 MSComm1.CommPort = COMPortNummer

 On Error Resume Next

 MSComm1.PortOpen = True
 If Err = 0 Then

  PortTest = True
  MSComm1.PortOpen = False
 Else
  PortTest = False
  MSComm1.PortOpen = False
 End If
End Function

Sub Port_Sına()
Dim Port_Sayisi As Integer

  Seyir.Text = Seyir.Text & Str(Time) & _
       " +++ COM portlar test edildi" & vbCrLf

  ComboCom.Clear

  For Port_Sayisi = 1 To 4
   If PortTest(Port_Sayisi) Then
    ComboCom.AddItem "COM" & Str(Port_Sayisi)
   End If
  Next

  If ComboCom.ListCount = 0 Then
   ComboCom.AddItem "COM Port Hazır Değil"
   Seyir.Text = Seyir.Text & Str(Time) & _
        " +++ COM Port Hazır Değil" & vbCrLf
        Else
```

```
  Seyir.Text = Seyir.Text & Str(Time) & _
       " +++" & Str(ComboCom.ListCount) & " COM Port Hazır" &
 End If

 ComboCom.ListIndex = 0
End Sub

Private Sub CmdAc_Click()
 Dim CPort As Byte
 Dim Settings As String

 Port_Sına

 Seyir.Text = Seyir.Text & Str(Time) & _
       " +++ COM Port açıldı" & vbCrLf

 If PortOpen = True Then
  MsgBox "COM Port zaten açık "

  Exit Sub
 End If

 CPort = Val(Mid(ComboCom.List(ComboCom.ListIndex), 4, _
       Len(ComboCom.List(ComboCom.ListIndex))))

 If CPort = 0 Then
  MsgBox "Hata! COM Port mevcut değil"

  Seyir.Text = Seyir.Text & Str(Time) & _
       " +++ COM Port açılamıyor " & vbCrLf
 Else
  MSComm1.CommPort = CPort
  Settings = ComboBaud.List(ComboBaud.ListIndex)
  Select Case ComboEslik.ListIndex
   Case 0: Settings = Settings & ",E"
   Case 1: Settings = Settings & ",M"
   Case 2: Settings = Settings & ",N"
   Case 3: Settings = Settings & ",O"
```

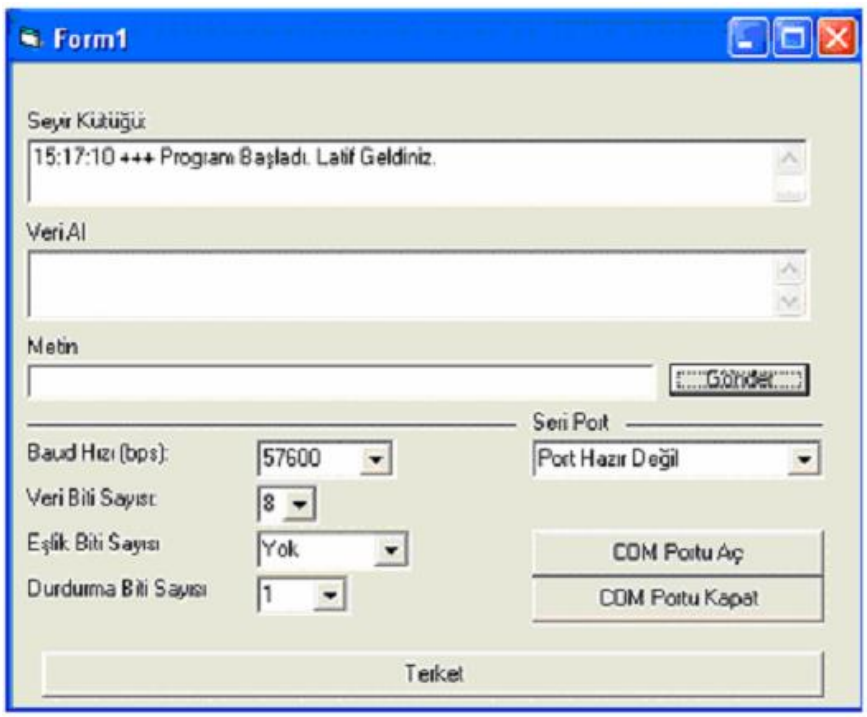
```
   Case 4: Settings = Settings & ",S"
  End Select

  Settings = Settings & "," & ComboVeri.List(ComboVeri.ListIndex)

  Settings = Settings & "," & ComboDur.List(ComboDur.ListIndex)

  MSComm1.Settings = Settings

  On Error Resume Next

  MSComm1.PortOpen = True

  If Err <> 0 Then
   MsgBox "Hata! COM Port mevcut değil"

   Seyir.Text = Seyir.Text & Str(Time) & " +++ COM Port açılamıyor " _
        & vbCrLf
  Else
   MSComm1.RThreshold = 1
   MSComm1.SThreshold = 1
   MSComm1.InputLen = 0

   PortOpen = True
  End If
 End If

End Sub

Private Sub cmdGonder_Click()
 If PortOpen = False Then
  MsgBox "Lütfen önceden portu açınız"
  Seyir.Text = Seyir.Text & Str(Time) & _
       " +++ Gönderim esnasında hata oluştu, Port " & _
       "Port açık değil" & vbCrLf
  Exit Sub
 End If

 If Gonder.Text = "" Then
```

```
 MsgBox "Bir metin yazınız"
 Seyir.Text = Seyir.Text & Str(Time) & " +++ Metin Yok" & _
      vbCrLf
 Exit Sub
End If

On Error Resume Next

MSComm1.Output = Gonder.Text + vbCrLf

If Err <> 0 Then
 MsgBox "Metin gönderilemedi!"
 Seyir.Text = Seyir.Text & Str(Time) & " +++ metin yok " & vbCrLf

Else
 Seyir.Text = Seyir.Text & Str(Time) & _
      " +++ """ + Gonder.Text & " "" gönderildi" & vbCrLf

 End If
Gonder.Text = ""
End Sub

Private Sub Cmdkapat_Click()
Seyir.Text = Seyir.Text & Str(Time) & _
      " +++ Port kapatılacak" & vbCrLf

 If PortOpen = False Then
  MsgBox "Açık port yok "


 Seyir.Text = Seyir.Text & Str(Time) & _
      " +++ Port kapatılamıyor " & vbCrLf
      Else
 On Error Resume Next

 MSComm1.PortOpen = False
 If Err <> 0 Then
  MsgBox "Port kapatma esnasında hata oluştu"
  Seyir.Text = Seyir.Text & _
```

```
        Str(Time) & " +++ COM Port kapatılamadı
  Else
   Seyir.Text = Seyir.Text & Str(Time) & " +++ COM
   PortOpen = False
  End If
 End If
End Sub
Private Sub cmdTerket_Click()
End
End Sub

Private Sub Form_Load()

 ComboCom.Clear
 ComboCom.AddItem "Port Hazır Değil"
 ComboCom.ListIndex = 0

 ComboBaud.AddItem "4800"
 ComboBaud.AddItem "9600"
 ComboBaud.AddItem "19200"
 ComboBaud.AddItem "38400"
 ComboBaud.AddItem "57600"
 ComboBaud.AddItem "115200"
 ComboBaud.ListIndex = 4

 ComboVeri.AddItem "6"
 ComboVeri.AddItem "7"
 ComboVeri.AddItem "8"
 ComboVeri.ListIndex = 2

 ComboEslik.AddItem "Tek"
 ComboEslik.AddItem "Çift"
 ComboEslik.AddItem "Yok"
 ComboEslik.AddItem "Mark"
 ComboEslik.AddItem "Space"
 ComboEslik.ListIndex = 2

 ComboDur.AddItem "1"
 ComboDur.AddItem "1.5"
```

```
ComboDur.AddItem "2"
ComboDur.ListIndex = 0

PortOpen = False

Seyir.Text = Str(Time) & _
     " +++ Program Başladı. Latif Geldiniz." & vbCrLf

End Sub

Private Sub MSComm1_OnComm()
Select Case MSComm1.CommEvent
  Case comEventOverrun:  MsgBox "Veri Kayboldu!"
  Case comEvReceive: Al.Text = Al.Text + MSComm1.Input
 End Select

End Sub
```

Program çalıştırıldığındaki ilk ekran görüntüsü:

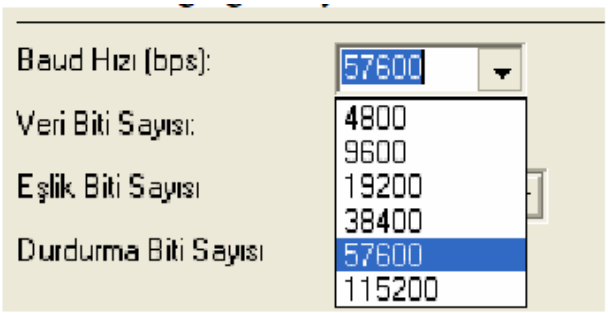


Şekil 3.8: Ekran görüntüsü

"COM Portu Aç" düğmesine tıklanarak seri port kullanıma açılır.

Şekil 3.9: Programın çalışması

Artık seri port parametreleri isteğe göre ayarlanır.

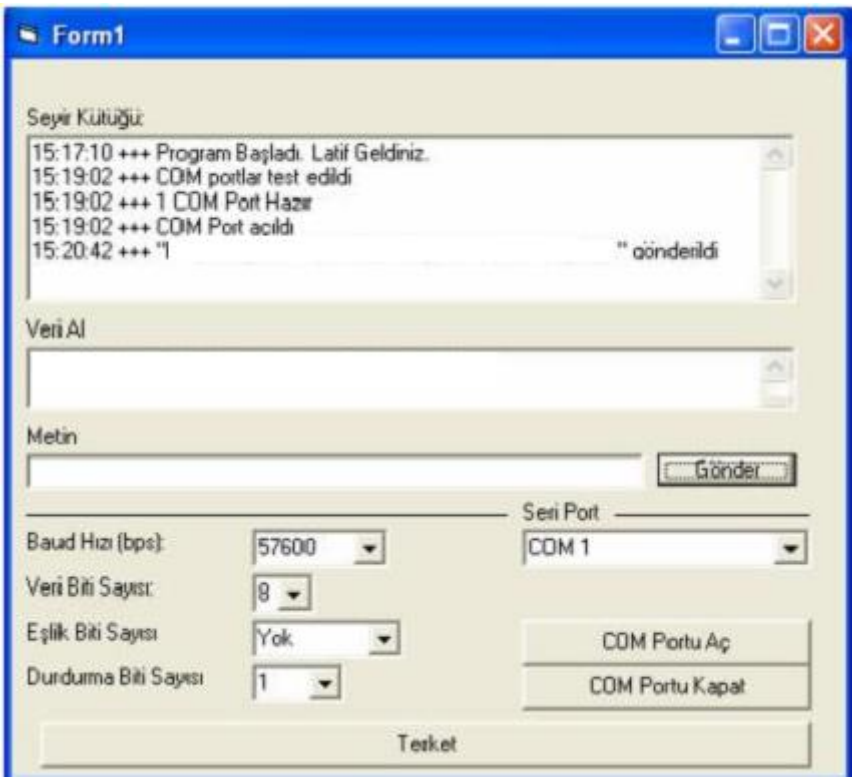


Şekil 3.10: Programın çalışması

İstenen veri hızı yazılır ve düğmeye tıklanır.

Şekil 3.11: Programın çalışması

Görüldüğü gibi yazılan mesaj geri geldi ve seyir günlüğüne kaydedildi.



Şekil 3.12: Programın çalışması

UYGULAMA FAALİYETİ

Aşağıdaki sorulara ilişkin uygulama faaliyetini yapınız.

- Örnek 3.2'de yapılan uygulamaya RTS-CTS el sıkışma sinyallerini ekleyin
- Dil kullanrak basit bir mesaj gönderme alma programı yapınız.
- Aynı dll programını ReadFile ve WriteFile API fonksiyonları ile yapınız.

İşlem Basamakları	Öneriler
➢ Form tasarımını yapınız. ➢ Bileşenlere uygun isimler veri-niz. ➢ Programı yazınız. ➢ Yazdığınız programı derleyiniz. ➢ Programda hata var ise bunları düzeltiniz. ➢ Ekran görüntüsünü kontrol edi-niz.	➢ Programda kullanacağınız değişken-lerin tipini belirlemeliniz. ➢ Değişken isimlendirme kurallarına dikkat etmelisiniz. ➢ Program satırlarının düzenli olmasına özen göstermelisiniz. ➢ Karar ifadelerinin belirlenen şartlara uygun olmasına dikkat etmelisiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. Aşağıdaki ifadelerden hangisi bir el sıkışma bağlantısı değildir?
A) CTS-RTS
B) DSR-DTR
C) DSR-DTR-DCD
D) CTS-DTR

2. Aşağıdakilerden hangisi bir yazılım el sıkışma anlaşmasıdır?
A) XON-XOFF
B) DSR-DTR
C) DSR-DTR
D) CTS-DTR
E) RxD-DTR

3. Aşağıdakilerden hangisi standart bir baud oranı değildir?
A) 19200
B) 110
C) 600
D) 3800
E) 115200

4. Veri bitini ayarlarken aşağıdakilerden hangisinin belirtilme zorunluluğu yoktur?
A) Stop biti
B) Eşlik biti
C) Başlangıç biti
D) Veri biti
E) Hiçbiri

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise Modül Değerlendirmeye geçiniz.

MODÜL DEĞERLENDİRME

Modülde yaptığınız uygulamaları tekrar yapınız.

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri Evet, kazanamadığınız becerileri Hayır kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Bileşenlere doğru isimler verdiniz mi?		
2. Değişken tanımlamalarını tam yaptınız mı?		
3. Olayları yerinde seçtiniz mi?		
4. Bileşen özelliklerini uygun atamalar yaptınız mı?		
5. For döngüsünü kullandınız mı?		
6. Şartlı ifadeler kullandınız mı?		
7. Dll dosyasını tanımladınız mı?		
8. API tanımlamalarını yaptınız mı?		
9. Program yazım işlemlerinin doğruluğunu kontrol		

DEĞERLENDİRME

Değerlendirme sonunda "Hayır" şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız "Evet" ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	**C**
2	**A**
3	**A**
4	**B**
5	**D**

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	**D**
2	**A**
3	**B**
4	**A**
5	**C**

ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1	**D**
2	**A**
3	**C**
4	**B**

KAYNAKÇA

- David I. Schneider, **Computer Programming Concepts and Visual Basic**, Pearson Custom Publishing, 1999.
- Jan Axelson, **Her Yönüyle Seri Port,** Bileşim Yayıncılık, İstanbul, 2000.
- Peter Norton, **Peter Norton's Guide to Visual Basic 6**, Macmillan Computer Publishing, 1998.
- Steven Holzner The Coriolis Group, **Visual Basic 6 Black Book**, 1998.
- Zeydin PALA, İhsan KARAGÜLLE, **Visual Basic 6.0 Pro**, Türkmen Yayınevi, İstanbul, 2002.